PRINCIPLES OF DIGITAL LOGIC

NAVAL EDUCATION AND TRAINING COMMAND

NAVEDTRA 106-01-00-79

Although the words "he", "him", and "his", are used sparingly in this manual to enhance communication, they are not intended to be gender driven nor to affront or discriminate against anyone reading *PRINCIPLES OF DIGITAL LOGIC*, NAVEDTRA 106-01-00-79.

PREFACE

This training manual is intended for use by U. S. Navy and Naval Reserve personnel whose duties require a general knowledge of logic circuits, Boolean algebra, and number systems.

This course is a prerequisite to NAVEDTRA 10088-B, *Digital Computer Basics* and MUST be completed prior to attempting NET 10088-B.

A basic background in transistors, such as that provided by NAVEDTRA 10087-C, *Basic Electronics,* or the *Navy Electricity-Electronics Training Series (NEETS)* is highly recommended as a prerequisite.

The principles of logic circuits, using switching logic, transistor logic, and truth tables, provides a basic approach to the logic used in computers. Boolean algebra and number systems provide the background necessary to enter the computer field.

This training manual was prepared by the Naval Education and Training Program Development Center, Pensacola, FL, for the Chief of Naval Education and Training.

1979

Stock Ordering No.
0507-LP-106-0100

Published by
NAVAL EDUCATION AND TRAINING PROGRAM
DEVELOPMENT CENTER

UNITED STATES
GOVERNMENT PRINTING OFFICE
WASHINGTON, D.C.: 1979

THE UNITED STATES NAVY

GUARDIAN OF OUR COUNTRY

The United States Navy is responsible for maintaining control of the sea and is a ready force on watch at home and overseas, capable of strong action to preserve the peace or of instant offensive action to win in war.

It is upon the maintenance of this control that our country's glorious future depends; the United States Navy exists to make it so.

WE SERVE WITH HONOR

Tradition, valor, and victory are the Navy's heritage from the past. To these may be added dedication, discipline, and vigilance as the watchwords of the present and the future.

At home or on distant stations we serve with pride, confident in the respect of our country, our shipmates, and our families.

Our responsibilities sober us; our adversities strengthen us.

Service to God and Country is our special privilege. We serve with honor.

THE FUTURE OF THE NAVY

The Navy will always employ new weapons, new techniques, and greater power to protect and defend the United States on the sea, under the sea, and in the air.

Now and in the future, control of the sea gives the United States her greatest advantage for the maintenance of peace and for victory in war.

Mobility, surprise, dispersal, and offensive power are the keynotes of the new Navy. The roots of the Navy lie in a strong belief in the future, in continued dedication to our tasks, and in reflection on our heritage from the past.

Never have our opportunities and our responsibilities been greater.

CONTENTS

NOTE

A question (or questions) follows each group of paragraphs. The questions are designed to determine if you understand the immediately preceding information. As you study, write out your answers to each question on a sheet of paper. If you have difficulty in phrasing an answer, restudy the applicable paragraphs. Do not advance to the next block of paragraphs until you are satisfied that you have written a correct answer.

When you have completed study of the text matter and written satisfactory answers to all questions on two facing pages of the book, compare your answers with those at the top of the next even-numbered page. If the answers match, you may continue your study with reasonable assurance that you have understood and can apply the material you have studied. Whenever your answers are incorrect, restudy the applicable material to determine why the book answer is correct and yours is not. If you make an honest effort to follow these instructions, you will have achieved the maximum learning benefits from each study assignment.

CHAPTER 1

LOGIC CIRCUITS

Rapid advances in computer technology have been made in the last two decades. Scientific and industrial progress undreamed of just a few years ago has become a reality. This is directly due to the speed and versatility of electronic digital computers and associated digital equipment. These digital devices process information today at a very fast rate. The principal control element of these devices is the digital logic circuit.

Logic circuits are so named because their operation is described by simple equations of a specialized logic algebra. Electronic gates make up logic circuits. These gates have one or more inputs and an output that depends on the combination(s) of their inputs. Nearly all the logic circuits found in today's equipment are formed from three basic circuit types: the AND gate, the OR gate, and the Inverter or Negator gate.

This chapter provides an introduction to the electronic and logical operation of the major logic circuits. It discusses the logic algebra used with logic circuits. Chapter 3 will describe the various number systems being used with digital equipment.

LOGIC COMPUTATION

The logic algebra used in the design, operation, and maintenance of the computer is called Boolean algebra. It is deceptively simple because each equation variable can have only one of two possible values. These two values are frequently indicated by the numbers 1 and 0, the terms high and low, or the statements true and false. For instance, in the equation A = f, if the variable A is a high, true, or 1, then the opposite value $\overline{A}$ (written as $\overline{A}$ and spoken as A NOT) would be termed low, false, or 0. If the reverse were the condition and A were low, false, or 0, then $\overline{A}$ would be termed a high, true, or 1. The important point to remember is that the variable (A, $\overline{A}$, B, $\overline{B}$, C, $\overline{C}$, etc.) can be only one of the two values at any given time.

Q1-1. How many different values can a single variable assume at any given time?

In ordinary algebra there are four basic arithmetic operations: addition, subtraction, multiplication, and division. In Boolean algebra three logical operations are used: AND, OR, and NOT. The AND operation requires that ALL equation variables be in their true condition at the same time for the function (f) of the equation to be true. A dot (·) between variables (A · B · C) or no dot indicator (ABC) specifies an AND operation. For the equation f = A · B · C (f = ABC) to be true, all three logic variables—A and B and C—must be true at the same time.

In the OR operation AT LEAST ONE of the logic variables must be in its true condition for the equation to be true. A plus sign (+) between variables (f = A + B + C) indicates the OR operation.

The Boolean NOT operation is indicated by a bar over the variable ($\overline{A}$ read as A NOT). It denotes the complement of a variable. For example, if A were true, then its complement

A1-1. One

$\overline{A}$, would be false; or if A were false, then its complement, $\overline{A}$, would be true.

Q1-2. What Boolean algebra operation is indicated by a plus sign (+)?

Two of the three Boolean operations just described have been converted into logic circuits that bear the same name, AND and OR. The third function, NOT, is performed by an Inverter or Negator circuit. There are additional logic circuits such as the NAND, NOR, and Exclusive OR which are combinations of the AND, OR, and Inverter circuits. Each of these circuits will be described in detail in this chapter. The electronic operation, a switch circuit analogy, and the current symbology used for logic diagrams will be included.

Using these logic circuits to develop a network that will perform the desired function of a Boolean equation is termed "implementing" or "mechanizing" the function. Another part of this publication will describe some of the math techniques for simplifying complex Boolean equations. This provides the means to determine the fewest possible logic circuits necessary to implement the equation. These simplification techniques are extremely useful in understanding the equipment operation.

LOGIC POLARITY

Any two distinct voltages can be used to represent the two logic possibilities—high and low, 1 and 0, or true and false. For example, a negative voltage could indicate a logic 0 and a positive voltage could indicate a logic 1. By reversing the logic, the opposite can be achieved—a negative voltage becomes a 1 and a positive voltage becomes a 0. Pulses can also be used, such as a negative pulse indicating a low (false or 0), and a positive pulse indicating a high (true or 1). A low could be indicated by the absence of a pulse, in which case the presence of a pulse would signify a high. As you can see, many combinations of logic expressions are possible and they may be used interchangeably. Most present logic systems use voltage polarity to define the variable condition, since positive and negative voltages are easily obtained and manipulated regardless of how the actual logic circuit operates.

Logic circuits are generally divided into two broad classes according to their polarity—positive logic and negative logic. Mixed positive and negative logic generally is not used because of the high cost of implementation. As used on logic diagrams, a signal or variable may assume either the active (true, high, 1) state or the inactive (false, low, 0) state. The voltage levels used and a statement indicating the use of positive or negative logic will usually be specified on logic diagrams supplied by the manufacturers.

In practice, many variations of logic polarity are used: for example, from a high-positive to a low-positive voltage, or from positive to ground; or, from a high-negative to a low-negative voltage, or from negative to ground. A brief discussion of the two general classes of polarity is presented in the following paragraphs.

Positive Logic

Positive logic polarity is defined as follows: when the signal that activates the circuit (a 1, high, or true) has an electrical level that is relatively more POSITIVE than the other logic state, the logic polarity is considered to be POSITIVE. The following examples show the manner in which positive logic may be employed.

EXAMPLE 1—Active signal—1, high, true = +10 volts

0, low, false = 0 volts

EXAMPLE 2—Active signal—1, high, true = 0 volts

0, low, false = -10 volts

In both examples the high state is always more positive than the low state, even though in example 2 the low state is a negative voltage.

Negative Logic

As you might suspect, NEGATIVE LOGIC is the reverse of positive logic and is defined as: when the signal that activates the circuit (a 1, high, or true) has an electrical level that is relatively more NEGATIVE than the other logic state, the logic polarity is considered to be NEGATIVE. Two of the possibilities for negative logic are shown in the following examples:

EXAMPLE 1—Active signal—1, high, true = 0 volts

0, low, false = +10 volts

EXAMPLE 2—Active signal—1, high, true = -10 volts

0, low, false = 0 volts

In the examples shown, the logic 1 state is always more negative than the logic 0 state, even though in example 1 both states are in the positive region.

Implementing positive or negative logic for digital equipment is a problem to be solved by design engineers. The difficulty for the technician in this area is limited to understanding the type of logic being used and keeping it in mind when troubleshooting.

The difference in circuit design is basically one of transistor type selection, PNP or NPN, and suitable polarity bias connections. Positive logic circuits will be described in detail in this chapter. Comparable negative logic circuits have been left for the reader to investigate through other sources such as the *Electronics Information and Maintenance Book.*

Q1-3. A logic circuit uses a high value negative voltage and a low value negative voltage as the input logic levels. The circuit is activated by the low value voltage. What is the logic polarity of the circuit?

THE AND OPERATION

Figure 1-1 shows the AND operation using toggle switches. Applying the Boolean principles

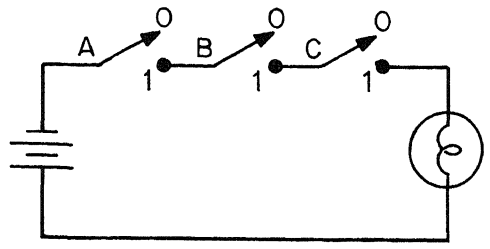


Figure 1-1.—Mechanical switch AND gate.

discussed previously, you will notice that ALL the switches must be closed for the circuit's bulb to be lighted. If any one switch is open, the bulb will not light. The truth table in figure 1-2 is a tabular means of evaluating a logical expression by showing all possible combinations of the input variables and the resultant output. This can also be described as a method of showing the relation of all output logic levels of a circuit to all possible combinations of input logic levels. As shown in the AND truth table, the only time the output is a logic 1 is when each input is a logic 1.

Q1-4. What must be the condition of the inputs to an AND gate for the output to be true?

The AND operation is one of the logic functions that can be mechanized with only

A	B	C	f = ABC
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Figure 1-2.—AND gate truth table.

A1-2. OR

A1-3. Positive

A1-4. All inputs must be true or a logic 1.

---

resistors and diodes. A representative circuit is shown in figure 1-3. For this circuit to work properly as an AND gate, certain conditions must be established. The first condition is, if this is a positive logic circuit, the high must be the more positive voltage level and assigned a logic value of 1. A more negative voltage level is used for the low and is assigned the logic value 0. Thus, if it is assumed that a high is equal to a positive voltage and a low is equal to a negative voltage and +V is equal to a larger positive voltage, one of the following circuit conditions will exist:

1. With a low (negative voltage) applied at A, B, and C ($\overline{A}\overline{B}\overline{C}$), a low will appear at f. This is true because under these conditions the diodes, CR1, CR2, and CR3, will be forward biased (current flow is against the point of the arrow), permitting a large amount of current to flow in the circuit. Since this large amount of current must flow through R, the IR voltage drop across R will be nearly equal to the potential difference between the input voltage and the supply voltage. Therefore, the voltage at the junction of the diodes and resistor (f output) will be a negative voltage with respect to ground, or a low.

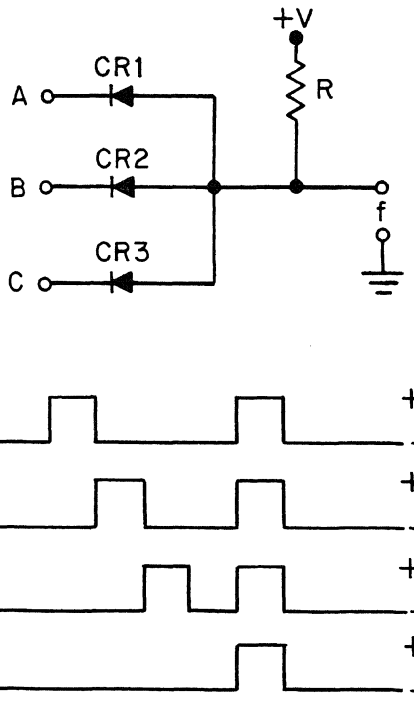


Figure 1-3.—Positive logic diode AND gate.

2. With an input of $A\overline{B}\overline{C}$ (a high on A, lows on B and C), a low will appear at f. This happens because the lows at B and C cause CR2 and CR3 to be forward biased. With CR2 and CR3 forward biased, a large amount of current will flow through CR2 and CR3, and the IR drop across CR2 and CR3 will be negligible. With only a slight IR drop across CR2 and CR3 (the major IR drop is across R as in 1 above), the lows at A and B will appear at the junction of the resistor and diodes. Since it now has a low on its anode and a high on its cathode, CR1 will be reverse biased. With CR1 reverse biased, little or no current will be permitted to flow in either direction through it. Therefore, the voltage at A (a high) will have little or no effect on the output signal.

Q1-5. Refer to figure 1-3. With a low or negative voltage applied to any one or all of the inputs the voltage drop across R will be such that the output will be a _____.

3. With highs at B and C and a low at A ($\overline{A}BC$), a low will appear at f. This is because under these conditions circuit operation is essentially the same as in example 2. The difference is that now CR1 will be forward biased and CR2 and CR3 will be reverse biased. Thus, the voltage at B and C will have little effect on the output signal.

4. If the inputs are all activated simultaneously by a high or positive voltage (ABC), the cathodes become more positive than

the anodes and the diodes cease conducting. When conduction stops, the voltage drop across R no longer exists and output f rises to the full value of the supply voltage. Thus, a high output exists only when ALL inputs are high.

The waveforms shown in figure 1-3 illustrate the relationship of the inputs to time. Note that the f waveform is low when any input waveform is low. When all inputs are high at the same time, the output waveform will also be high.

Figure 1-4 is a schematic diagram of a representative three-input positive logic transistor AND gate. Although the purpose of the circuit is exactly the same as the diode AND gate, it provides gain and has a faster operating speed. The waveforms from figure 1-3 and the truth table from figure 1-2 also apply to the transistor circuit.

In this circuit, both collector and emitter are reverse biased and therefore are at cutoff. Under these conditions no collector current flows and the output voltage will be equal to the collector supply voltage. Assuming that proper circuit parameters have been chosen, a high signal (or logic 1) applied to any one or a combination of the inputs, but NOT ALL, will cause a certain amount of current to flow through R4. This current causes an IR voltage drop across R4 which opposes the negative bias applied to the emitter. However, this IR voltage drop will not be sufficient to overcome the negative bias; hence, the emitter and collector remain cut off and there is little or no change in output voltage.

With a high applied at all inputs, sufficient current now flows through R4 so that the resulting IR voltage drop overcomes the negative bias and the emitter is forward biased. With the emitter forward biased and collector current flow through R5, the IR voltage drop will cause the output voltage to swing in a positive direction (become less negative) and provide a high (logic 1) at the output. It will remain at a relatively positive level until any of the input signals are dropped to a low (logic 0). When any one or all of the input signals are removed the transistor Q1 is again cut off, the collector current drops to zero, and the output voltage returns to the $-V_{CC}$ level. R1, R2, and R3 have high resistance values to provide isolation between inputs.

Q1-6. Refer to figure 1-4. With a high (logic 1) applied to all three inputs, the positive voltage on the base/emitter junction of the transistor will be above _____.

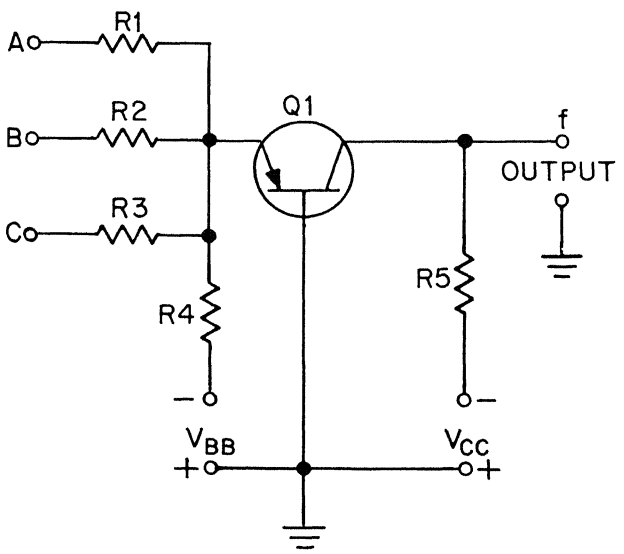


Figure 1-4.—Positive logic transistor AND gate.

The currently used symbol for a positive logic AND gate is shown in figure 1-5. Keep in mind that the variable inputs (A, B, and C) to the symbol are the same as the inputs to the transistor circuit shown in figure 1-4. All the logic diagram symbols used for explanation in this publication will be from the latest available American National Standard publication *Graphic Symbols for Logic Diagrams*, ANSI Y32.14-1973.

Up to now only the positive logic AND gate has been discussed. Recall from earlier in this

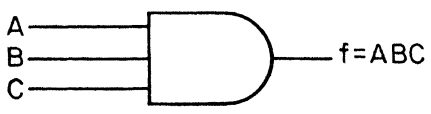


Figure 1-5.—Positive logic AND gate symbol.

A1-5. low

A1-6. cutoff

chapter that both positive and negative logic circuits may be employed in a piece of equipment. Some method must be used to identify the polarity of the logic being used. Figure 1-6 shows the current method of illustrating negative logic. The symbol is the same as for positive logic with the addition of small right triangles, called flags, placed on the input and output lines. Remember the difference between negative and positive logic. Positive logic states that the activating signal for a circuit is of a relatively more positive voltage level, while negative logic uses a relatively more negative voltage level to activate the logic circuit. The logical function of any circuit remains the same for either positive or negative logic; an AND gate still requires all inputs to be true for the output to be true.

Obviously, since the two logic types (negative and positive) are using different polarity voltages, the electronic operation of the circuit itself must be basically reversed. However, the logical operation is the same. It cannot be emphasized strongly enough that the right triangles DO NOT indicate a low, false, or 0 logic level. All they show is that the signal that enables the circuit is of a relatively more negative voltage level.

Q1-7. What is indicated by the addition of flags to a logic symbol?

THE OR OPERATION

An OR gate mechanized with toggle switches is shown in figure 1-7. For the function to be

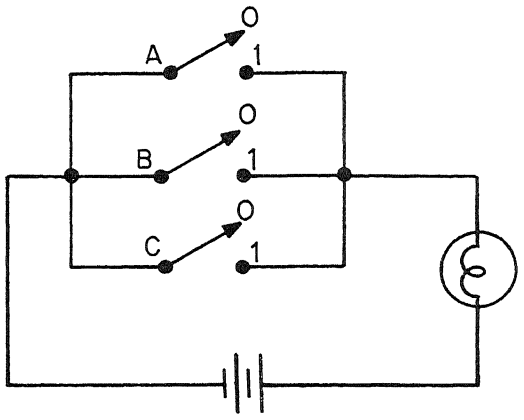


Figure 1-7.—Mechanical switch OR gate.

completed (the light bulb lighted) any single switch must be closed. If all the switches are left open, the light bulb will not light. This follows the Boolean OR principle stated earlier in this chapter. The OR gate truth table (fig. 1-8) shows that the circuit will output a logic 1 when any single switch or combination of switches is closed. A logic 0 level will be output only when all the switches are in the 0 position at the same time.

Q1-8. What must be the condition of the inputs to an OR gate for the output to be true?

As with the AND operation, the OR operation, f = A + B + C, may also be mechanized by using resistor diode logic. This is

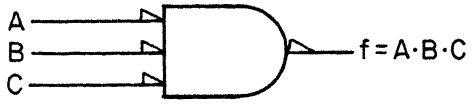


Figure 1-6.—Negative logic AND gate symbol.

A	B	C	f = A + B + C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Figure 1-8.—OR gate truth table.

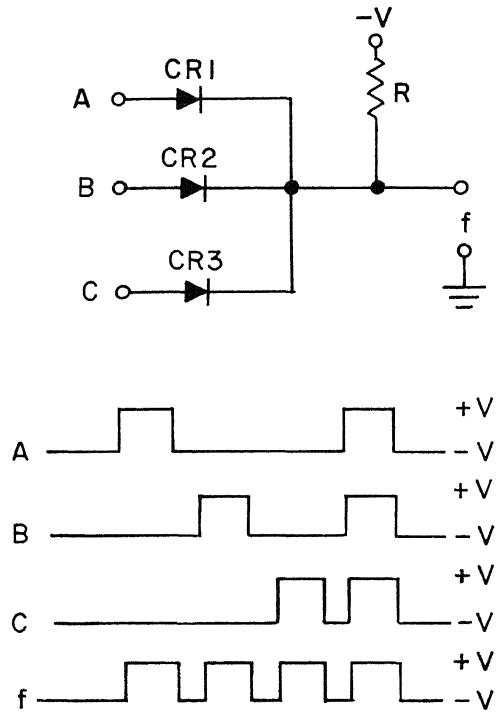


Figure 1-9.—Positive logic diode OR gate.

shown in figure 1-9. For this circuit to function as an OR gate certain conditions must again be established. A high, the more positive voltage, must be assigned a logic value of 1, and a low, the more negative voltage, must be assigned a logic value of 0. Thus, if it is assumed that a high is equal to a positive voltage and a low is equal to a negative voltage and -V is equal to a larger negative voltage, one of the following four circuit conditions will exist.

1. With lows applied at A, B, and C ($\overline{A} + \overline{B} + \overline{C}$), a low will appear at the output f. This is because, under these conditions, all three diodes, CR1, CR2, and CR3, are reverse biased and there will be no current flow through the circuit. With no current flow, there is no IR voltage drop across R. Therefore, the -V will appear at the output f, causing it to be a low.

2. If the input is $A + \overline{B} + \overline{C}$ (a high on A and lows on B and C), the output at f will be a high. This is because a high on A will forward bias CR1. With CR1 forward biased, a large amount of current will flow through CR1 and R. The IR voltage drop across CR1 will be negligible (the major IR drop is across R). The high on A will appear at the junction of the diodes and at the output f. Since they now have a low on their anodes and a high on their cathodes, CR2 and CR3 will be reverse biased and the voltage at the input of B or C will have little effect on the output voltage.

3. With a high applied at B and/or C and a low applied at A, a high will appear at f. This is because, under these conditions, circuit operation is essentially the same as in example 2. The difference is that CR2 and/or CR3 will be forward biased and CR1 will be reverse biased. Thus, the voltage at A now has little or no effect on the output voltage.

4. If highs are applied to all the inputs ($A + B + C$), a high will appear at the output f. This is due to the fact that circuit operation will be essentially the same as in examples 2 and 3. The difference is that all the diodes are forward biased resulting in current flow through each of them.

Q1-9. Refer to figure 1-9. If all the diodes are forward biased, the output will be a _____.

The schematic of a typical three-input positive logic transistor OR gate is shown in figure 1-10. As with the diode OR gate, the logic

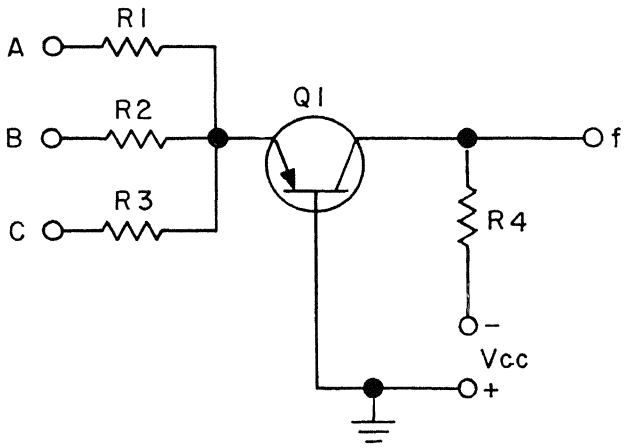


Figure 1-10.—Positive logic transistor OR gate.

A1-7. Negative logic is being used.

A1-8. Any one of the input must be a high (logic 1).

A1-9. high

---

level 1 (the more positive voltage) is a high. A low is the logic level 0 (the more negative voltage). In this circuit, the collector is reverse biased, while the emitter is left floating. Thus, with no signal applied at any input ($\overline{A} + \overline{B} + \overline{C}$) current cannot flow and the output voltage will be the same as V_{CC}, the negative collector supply voltage. A logic level 1, or high, applied to any or all of the inputs will forward bias the emitter. This will cause sufficient collector current to flow so that the IR voltage drop across R4 will be equal to the collector supply voltage. This causes the output voltage to go relatively positive and the output becomes a high (logic 1).

Q1-10. Refer to figure 1-10. With no signal applied to the inputs, the output f is equal to_____.

When the input signal is removed, the circuit returns to its quiescent state with no collector current flow and $-V_{CC}$, or a low, at the output. Therefore, a positive going input signal results in a positive going output signal. Since in its quiescent state the circuit is operating at cutoff, a negative going input signal will have little or no effect on circuit operation. The input resistors normally have high resistive values to isolate each input from every other input.

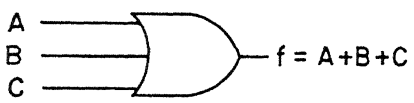


Figure 1-11.—Positive logic OR gate symbol.

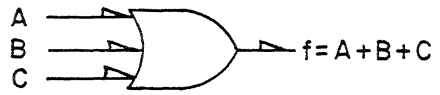


Figure 1-12.—Negative logic OR gate symbol.

Symbols for positive and negative logi gates are shown in figures 1-11 and respectively. Note again that the only diffe between the two is that the negative symbol has small flags on the input and ou lines. The three inputs, A, B, and C, are the for both symbols and the Boolean ou equation for each is the same, f = A + B + C

Q1-11. The output equation for a negative OR gate is_____.

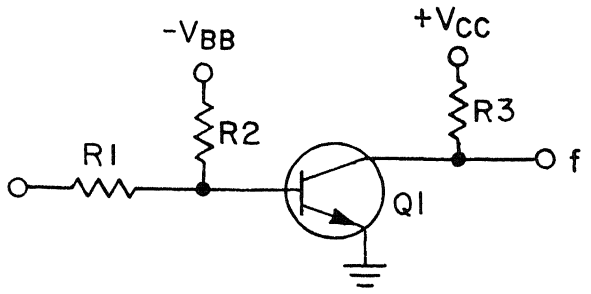
THE INVERTER GATE

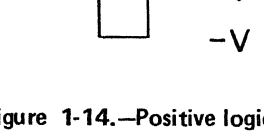
Another interesting and very useful Bo function is the NOT operation, performe the Inverter gate. Basically, it inverts its inp provide the opposite output. For example, input is high, the output will be low. Or, input is low, the output will be high. inversion is more commonly known in Bo algebra as negation or complementing. terms are equally valid. A truth table fo Inverter function is shown in figure 1-1 shows that the output is the complement o input.

Q1-12. If the input to an Inverter gate is output from the gate is_____.

A	f = $\overline{A}$
1	0
0	1

Figure 1-13.—Inverter gate truth table.





igure 1-14.—Positive logic Inverter gate and waveforms.

A schematic diagram for an Inverter or legator gate and the associated waveforms are nown in figure 1-14. Operating conditions for nis circuit are the same as for the AND and OR ates. A high or logic level 1 is the more positive oltage level and a logic 0 or low is the more egative level (positive logic).

With no signal input (a 0 or a low), the ransistor, acting as a switch, is cut off by the egative base voltage, V_{BB}. This reverse biases he transistor Q1, base-emitter junction. The utput is a positive voltage, V_{CC}, or a high ecause no collector current flows. A positive nput voltage (a 1 or a high) will forward bias he base-emitter junction. Collector current will low, which will cause an IR voltage drop across 23 nearly equal to V_{CC}. This will bring the utput to a logic 0 or low condition. Both nputs, therefore, satisfy the truth table equirements. The output will be the opposite ogic signal of the input.

Q1-13. Refer to figure 1-14. The voltage drop across R3 is nearly equal to V_{CC}. This indicates that the input signal is_____.

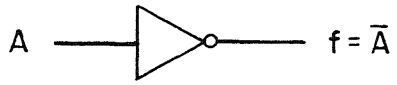


Figure 1-15.—Positive logic Inverter gate symbol.

Symbols for positive and negative logic Inverter gates are shown in figures 1-15 and 1-16, respectively. The large triangle is the standard symbol for an amplifier and the small circle is called a negation indicator. Both symbols are required to illustrate an Inverter gate used by itself to change logic levels between other logic elements. When attached to another logic symbol only the negation indicator is used. This point will be further clarified in the next logic circuit to be discussed. Note again (refer to fig. 1-15 and fig. 1-16) that the only difference is that the negative logic symbols use small right triangles attached to the input and output lines and the positive logic symbols do not.

Q1-14. How many symbols are required to indicate an Inverter gate used by itself?

THE NAND OPERATION

Combining the AND and Inverter logic circuits forms what is commonly called the NAND circuit (shortened from NOT-AND). Basically, the circuit operates by ANDing a number of logic signals together in a manner such as has been previously described. The output of the AND gate (fig. 1-13) is then used as the input to an Inverter circuit (fig. 1-14) which complements or negates the result of the

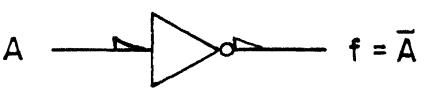


Figure 1-16.—Negative logic Inverter gate symbol.

A1-7. Negative logic is being used.

A1-8. Any one of the input must be a high (logic 1).

A1-9. high

---

level 1 (the more positive voltage) is a high. A low is the logic level 0 (the more negative voltage). In this circuit, the collector is reverse biased, while the emitter is left floating. Thus, with no signal applied at any input ($\overline{A} + \overline{B} + \overline{C}$) current cannot flow and the output voltage will be the same as V_{CC}, the negative collector supply voltage. A logic level 1, or high, applied to any or all of the inputs will forward bias the emitter. This will cause sufficient collector current to flow so that the IR voltage drop across R4 will be equal to the collector supply voltage. This causes the output voltage to go relatively positive and the output becomes a high (logic 1).

Q1-10. Refer to figure 1-10. With no signal applied to the inputs, the output f is equal to _____.

When the input signal is removed, the circuit returns to its quiescent state with no collector current flow and $-V_{CC}$, or a low, at the output. Therefore, a positive going input signal results in a positive going output signal. Since in its quiescent state the circuit is operating at cutoff, a negative going input signal will have little or no effect on circuit operation. The input resistors normally have high resistive values to isolate each input from every other input.

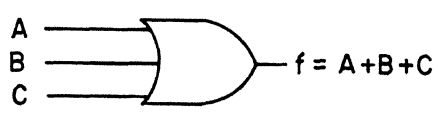


Figure 1-11.—Positive logic OR gate symbol.

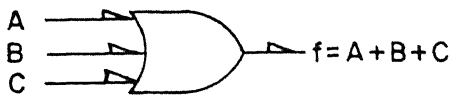


Figure 1-12.—Negative logic OR gate symbol.

Symbols for positive and negative logic OR gates are shown in figures 1-11 and 1-12, respectively. Note again that the only difference between the two is that the negative logic symbol has small flags on the input and output lines. The three inputs, A, B, and C, are the same for both symbols and the Boolean output equation for each is the same, f = A + B + C.

Q1-11. The output equation for a negative logic OR gate is_____.

THE INVERTER GATE

Another interesting and very useful Boolean function is the NOT operation, performed by the Inverter gate. Basically, it inverts its input to provide the opposite output. For example, if the input is high, the output will be low. Or, if the input is low, the output will be high. This inversion is more commonly known in Boolean algebra as negation or complementing. Both terms are equally valid. A truth table for the Inverter function is shown in figure 1-13. It shows that the output is the complement of the input.

Q1-12. If the input to an Inverter gate is Z, the output from the gate is_____.

A	$f = \overline{A}$
1	0
0	1

Figure 1-13.—Inverter gate truth table.

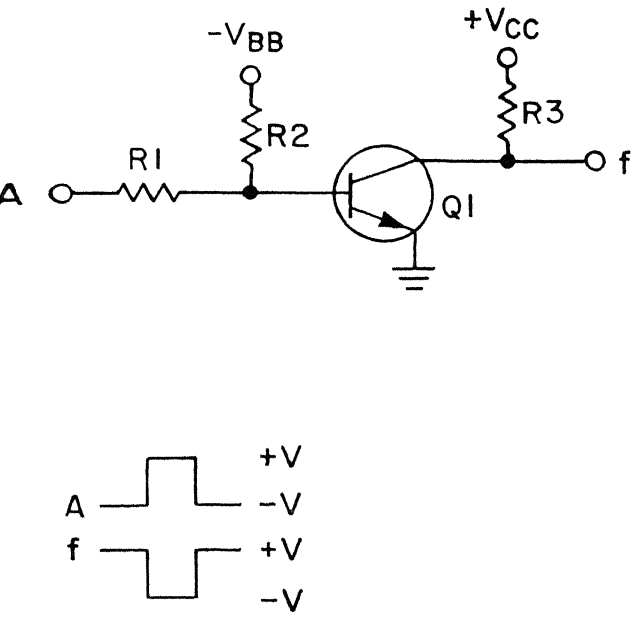


Figure 1-14.—Positive logic Inverter gate and waveforms.

A schematic diagram for an Inverter or Negator gate and the associated waveforms are shown in figure 1-14. Operating conditions for this circuit are the same as for the AND and OR gates. A high or logic level 1 is the more positive voltage level and a logic 0 or low is the more negative level (positive logic).

With no signal input (a 0 or a low), the transistor, acting as a switch, is cut off by the negative base voltage, V_{BB}. This reverse biases the transistor Q1, base-emitter junction. The output is a positive voltage, V_{CC}, or a high because no collector current flows. A positive input voltage (a 1 or a high) will forward bias the base-emitter junction. Collector current will flow, which will cause an IR voltage drop across R3 nearly equal to V_{CC}. This will bring the output to a logic 0 or low condition. Both inputs, therefore, satisfy the truth table requirements. The output will be the opposite logic signal of the input.

Q1-13. Refer to figure 1-14. The voltage drop across R3 is nearly equal to V_{CC}. This indicates that the input signal is_____.

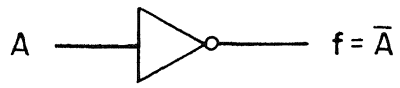


Figure 1-15.—Positive logic Inverter gate symbol.

Symbols for positive and negative logic Inverter gates are shown in figures 1-15 and 1-16, respectively. The large triangle is the standard symbol for an amplifier and the small circle is called a negation indicator. Both symbols are required to illustrate an Inverter gate used by itself to change logic levels between other logic elements. When attached to another logic symbol only the negation indicator is used. This point will be further clarified in the next logic circuit to be discussed. Note again (refer to fig. 1-15 and fig. 1-16) that the only difference is that the negative logic symbols use small right triangles attached to the input and output lines, and the positive logic symbols do not.

Q1-14. How many symbols are required to indicate an Inverter gate used by itself?

THE NAND OPERATION

Combining the AND and Inverter logic circuits forms what is commonly called the NAND circuit (shortened from NOT-AND). Basically, the circuit operates by ANDing a number of logic signals together in a manner such as has been previously described. The output of the AND gate (fig. 1-13) is then used as the input to an Inverter circuit (fig. 1-14) which complements or negates the result of the

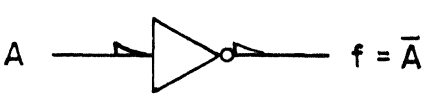


Figure 1-16.—Negative logic Inverter gate symbol.

A1-10. a low or a logical 0

A1-11. $f = A + B + C$

A1-12. $\overline{Z}$

A1-13. high or a 1

A1-14. Two; an amplifier and a negation indicator

AND operation. Keep in mind that the Inverter circuit output is the opposite of its input. If the input is A, the output is $\overline{A}$. Figure 1-17 shows the NAND truth tables. The only low output is caused by ALL high inputs.

Some new concepts which arise here are the complementing of arithmetic indicators, the ANDing sign, and the ORing sign. The complement of an AND operation, as might be suspected, is an OR, while the negation of an OR is an AND. For example, suppose the AND circuit output is $A \cdot B \cdot C$ and it is applied as the input to an Inverter circuit. With $A \cdot B \cdot C$ as its input, the Inverter circuit will output $\overline{A \cdot B \cdot C}$. The long bar across the variables and their algebra indicators is called a VINCULUM. The vinculum is a Boolean algebra indicator which requires EVERYTHING under it to be complemented or negated, including the algebra indicators. Therefore, $\overline{A \cdot B \cdot C}$ is equivalent to $\overline{A} + \overline{B} + \overline{C}$. The variables are complemented to their NOT form and the AND symbols are complemented to their NOT form of OR.

A	B	C	$f = \overline{ABC}$ (OR $\overline{A}+\overline{B}+\overline{C}$)
0	0	0	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

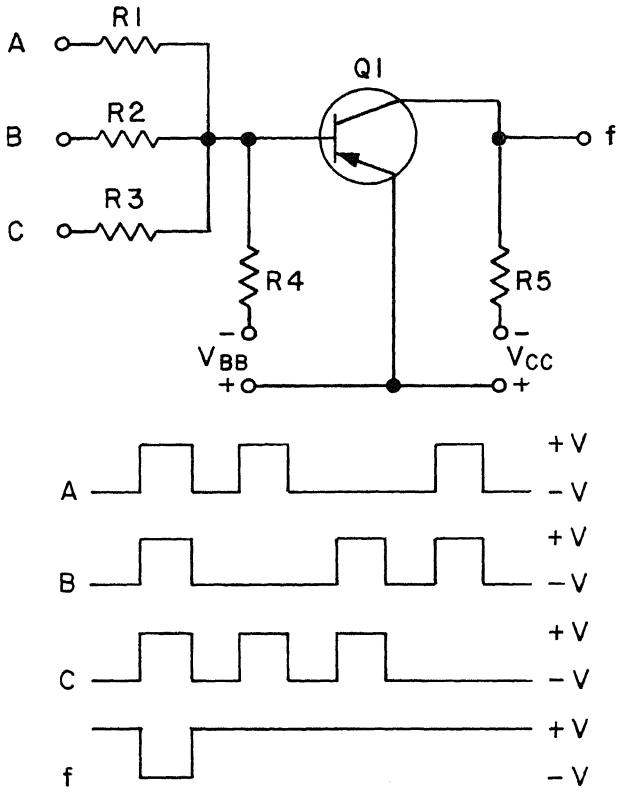


Figure 1-18.—Positive logic NAND gate and waveforms.

Q1-15. The vinculum indicates that _____.

Q1-16. What two logic circuits are combined to form a NAND gate?

Figure 1-18 is the schematic for a representative positive logic NAND gate. For this circuit to work properly as a NAND gate, certain conditions must be established. The first condition which must be met in a positive logic circuit is that the high or logic 1 state must be a

0 state. The negative base bias, V_{BB}, forward biases the base-emitter junction of Q1, causing the transistor to conduct heavily with no signals applied. A set of typical waveforms is included in figure 1-18. When a logic 0 or low is applied to all inputs, the base-emitter junction will still be forward biased. This causes transistor Q1 to conduct heavily and operate in the saturation region. The resulting heavy collector current flow through R5 produces an IR voltage drop across R5 which opposes the collector supply voltage, V_{CC}. This causes the collector voltage to go to a more positive voltage level, and produces a high at output f. Assuming the parameters have been properly chosen, a positive going signal at any one or two of the three inputs will cause a certain amount of current flow through R4. The current will cause an IR voltage drop across R4 which opposes the base bias supply voltage. This voltage drop will NOT be of sufficient magnitude to reduce the base-emitter bias voltage to zero and thereby cut off the transistor. The base-emitter junction will remain forward biased and transistor Q1 will continue to conduct heavily and remain in the saturation region. There will be no change in the output voltage, and a high or logic 1 will still appear as the output at f. With a positive going signal (logic 1) applied at each input simultaneously, enough current flows through R4 so that the IR voltage drop across R4 will be large enough to overcome the base bias voltage V_{BB}. This causes the base-emitter junction to be reverse biased and cuts off transistor Q1. Collector current will then cease and the IR voltage drop across R5 will be at a minimum. At this point the output voltage at f will be nearly equal to the collector supply voltage, V_{CC}, and a low or logic 0 will appear at the output.

Q1-17. Refer to figure 1-18. What is the condition of the emitter-base junction of Q1 with a positive signal on all three inputs?

Symbols for both positive and negative logic NAND gates are shown in figures 1-19 and 1-20,

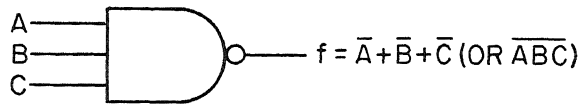


Figure 1-19.—Positive logic NAND gate symbol.

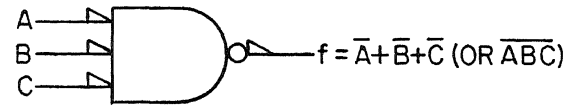


Figure 1-20.—Negative logic NAND gate symbol.

respectively. Note that the symbols are combinations of the AND gate and the Inverter circuit symbols. The output function for both symbols is the same, $f = \overline{ABC}$. The only difference is that the negative logic symbol uses the small right triangles on the input and output lines.

THE NOR OPERATION

The NOR gate combines the functions of an OR gate and the Inverter circuit. It is possibly the most common logic circuit in use today. Figure 1-21 is an illustration of the NOR truth table. The only true output occurs when all inputs are false.

Q1-18. What is the output of a NOR gate with each input a 0?

A	B	C	$f = \overline{A+B+C}$ (OR $\overline{A}\cdot\overline{B}\cdot\overline{C}$)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Figure 1-21.—NOR gate truth table.

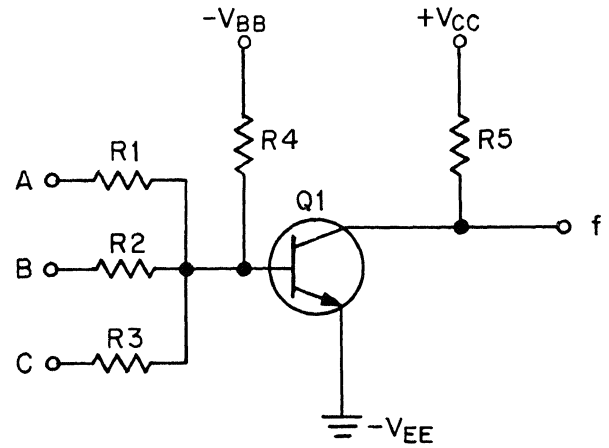
A1-15. everything under it must be complemented

A1-16. An AND and an Inverter

A1-17. It is reverse biased.

A1-18. True or a logic 1

Refer to the schematic diagram of the NOR circuit shown in figure 1-22 to determine initial conditions. First, this is a positive logic gate. The high or logic 1 is more positive than the low or logic 0. The low is a slightly more negative voltage. V_{BB}, the base bias supply, is more negative than V_{EE}, the emitter supply voltage.



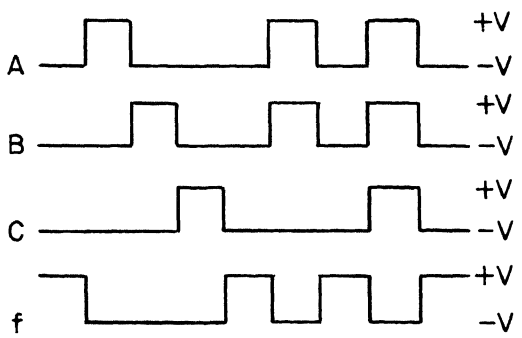


Figure 1-22.—Positive logic NOR gate and waveforms.

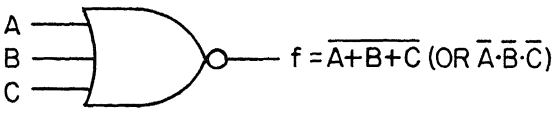


Figure 1-23.—Positive logic NOR gate symbol.

With no inputs at A or B or C ($\overline{A} + \overline{B} + \overline{C}$), the base-emitter junction is reverse biased and transistor Q1 is cut off. No collector current flows and the output f is essentially equal to $+V_{CC}$. The output is a high or logic 1. A positive input pulse on any one input or a combination of inputs will cause current to flow through R4. This will result in an IR voltage drop across R4 which opposes $-V_{BB}$. This will forward bias the base-emitter junction of transistor Q1 and it will conduct heavily. The resultant current flow through R5 to the $+V_{CC}$ supply causes an IR voltage drop across R5. This, in turn, causes the output at f to go in a relatively negative direction. The output will then be a low or logic 0. Any high (logic 1) in will provide a low (logic 0) out of the circuit. To obtain a high (logic 1) out of the circuit each input must be a low (logic 0).

Symbols for both positive and negative NOR gates are shown in figures 1-23 and 1-24, respectively. Note that the symbols are combinations of the OR gate and Inverter circuit symbols. The output of each is a complemented OR function $f = \overline{A + B + C}$, which may also be written as $f = \overline{A} \cdot \overline{B} \cdot \overline{C}$.

Q1-19. Refer to figure 1-22. With a logic 1 on two inputs the voltage level at point f is equal to _____.

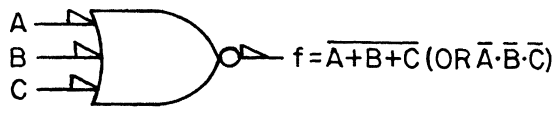


Figure 1-24.—Negative logic NOR gate symbol.

A	B	$f = A\bar{B} + \bar{A}B$
0	0	0
0	1	1
1	0	1
1	1	0

Figure 1-25.—Exclusive OR gate truth table.

THE EXCLUSIVE OR OPERATION

The truth table for an Exclusive OR gate is shown in figure 1-25. It shows that there is a true output from the gate when EITHER of the inputs is true BUT NOT BOTH at the same time ($f = A\bar{B} + \bar{A}B$). This circuit is normally used in the arithmetic section of a digital computer.

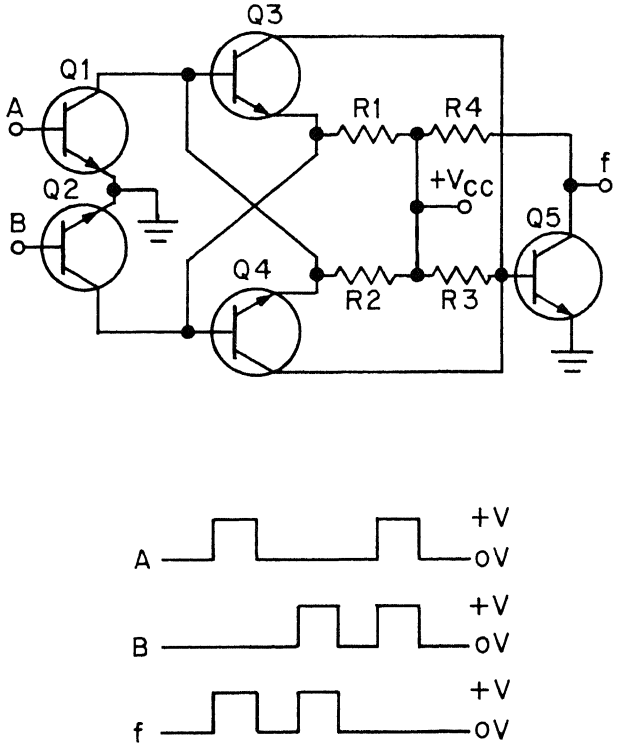


Figure 1-26.—Positive logic Exclusive OR gate and waveforms.

Q1-20. An Exclusive OR gate will have a true output when either of the inputs is true but not both at the same time. The Boolean equation that expresses this is _____.

Figure 1-26 depicts the schematic diagram for an Exclusive OR gate and its operating waveforms. Q1 and Q2 each form an inhibitor circuit while the other transistors are used as a single OR circuit. With no signal or zero volts applied at A and B, the emitter-base junction of Q1 and Q2 will be reverse biased. Since the transistors are reverse biased, little or no collector current will flow and the IR voltage drop across R1 and R2 will be negligible. Therefore, the voltages appearing at the bases and emitters of Q3 and Q4 will be nearly equal to V_{CC}. This will reverse bias the emitter-base junction of Q3 and Q4. Under these conditions Q3 and Q4 will be cut off and little or no collector current will flow. The IR voltage drop across R3 will be negligible and the voltage appearing at the base of Q5 will be nearly equal to V_{CC}. With this positive voltage applied to its base, the emitter junction of Q5 will be forward biased. Enough collector current will flow so as to produce an IR voltage drop across R4 (which is equal to and opposing V_{CC}). Therefore, the output f will be at or near zero units, as shown in the waveforms.

Q1-21. Refer to figure 1-26. With no signal applied to either input, what will be the condition of Q3 and Q4?

With a positive going signal applied at A and no signal applied at B, the Q1 emitter-base junction will be forward biased and the Q2 emitter-base junction will be reverse biased. Q1 will conduct heavily and Q2 will remain cut off. Q1 collector current, in flowing through R2, produces an IR voltage drop across R2 nearly equal to V_{CC}. Since this drop opposes V_{CC}, the Q4 emitter and Q3 base will be at or near zero

A1-19. a logic 0

A1-20. $f = A\overline{B} + \overline{A}B$

A1-21. Cutoff

volts. With zero volts on its base and a positive voltage on its emitter, Q3 remains cut off. Q4, on the other hand, will conduct heavily since it will have a positive voltage on its base, forward biasing its emitter-base junction. Q4 collector current, in flowing through R3, produces an IR voltage drop across R3 nearly equal to V_{CC}. Since this drop opposes V_{CC}, the base of Q5 will be at or near zero volts. With its emitter grounded and zero volts on its base, Q5 will be at or near cutoff. With little or no collector current flow in Q5, the IR voltage drop across R4 will be negligible, and the voltage at the output f will be nearly equal to V_{CC}. The output will be a high (logic 1).

With a positive signal at B and no signal at A the same sort of situation will exist. Only now, Q1 will be cut off and Q2 will be conducting. With Q1 cut off and Q2 conducting, Q4 will be cut off and Q3 will be conducting. This again reduces the voltage appearing at the base of Q5 to approximately zero volts, thus causing Q5 to be cut off. With Q5 cut off, the voltage appearing at its collector and, hence, the output will be nearly equal to V_{CC}. The output will be a high (logic 1).

Q1-22. Refer to figure 1-26. With a 1 on either of the inputs, but not both, what will be the condition of Q5?

A positive going signal at both inputs will cause Q1 and Q2 both to conduct heavily. This

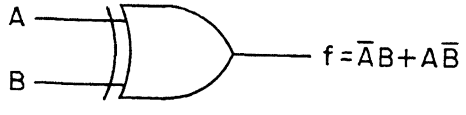


Figure 1-27.—Positive logic Exclusive OR gate symbol.

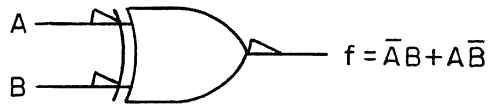


Figure 1-28.—Negative logic Exclusive OR gate symbol.

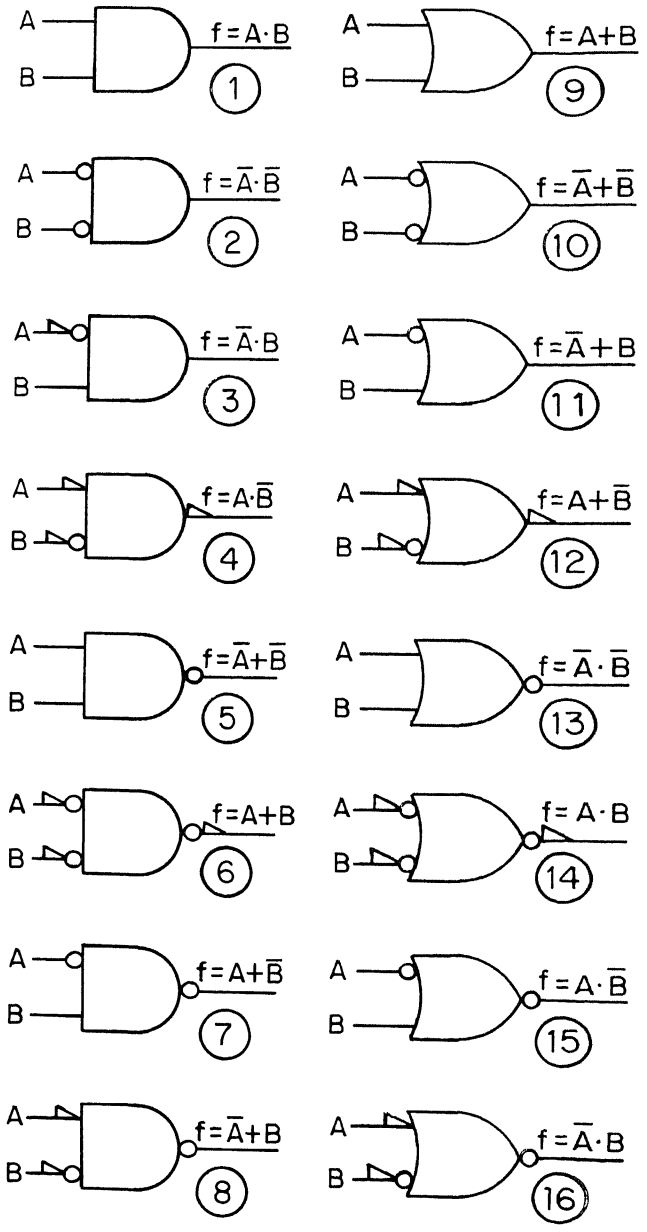


Figure 1-29.—Various logic circuit symbols and their functional outputs.

will result in the voltage on their collectors being reduced to approximately zero volts. Zero volts will also appear on the bases and emitters of Q3 and Q4 which will cause them (Q3 and Q4) to remain cut off. With Q3 and Q4 cut off, the IR voltage drop across R3 will be negligible and the voltage appearing at the base of Q5 will be almost equal to V_{CC}. With this positive voltage on its base, Q5 will conduct heavily and produce an IR voltage drop across R4 nearly equal to V_{CC}. Since this drop opposes V_{CC}, the output will be very close to zero volts.

Q1-23. Refer to figure 1-26. With a 1 applied to both inputs at the same time, why is the voltage at point f equal to zero volts?

Logic diagrams for positive and negative logic Exclusive OR gates appear in figures 1-27 and 1-28, respectively. Note that both symbols have the same output, $f = \overline{A}B + A\overline{B}$.

Logic Circuit Symbols

A number of logic circuit symbols and their functional outputs are shown in figure 1-29. It is readily apparent that each of the functional outputs can be mechanized by more than one circuit configuration, as shown by diagrams 1 and 14 in figure 1-29. The output of both of these symbols is $f = A \cdot B$, an AND function, even though diagram 14 is a NOR symbol with negated inputs and uses negative logic. Diagrams 5 and 10 have the same output, but one symbol is a NAND gate and the other is an OR gate with negated inputs. Both use positive logic. Keep in mind that the electronic operation of the logic circuit, whether it uses negative or positive logic and basically what type of logic circuit it is, makes no difference in the logical operation of a piece of equipment. The functional output of the symbol is all that counts. An equipment design engineer can determine what Boolean equation is necessary and a group of logic circuits will be selected to implement the desired function.

Q1-24. Refer to figure 1-29. What two symbols have the output A + B?

Q1-25. Refer to figure 1-29. What other symbol has the same functional output as symbol 16?

The basics of Boolean algebra, including techniques for developing logic diagrams from Boolean expressions and for simplifying complex equations, will be discussed in the next chapter of this book.

A1-22. At or near cutoff

A1-23. The heavy conduction of Q5 causes a voltage drop across R4 nearly equal to V_{CC} with the result that the voltage at point f is near or equal to zero.

A1-24. 6 and 9

A1-25. 3

CHAPTER 2

BOOLEAN ALGEBRA

In the mid-nineteenth century an English mathematician named George Boole presented a number of pamphlets on symbolic logic. The currently used theories concerning analysis of logical processes are based on these pamphlets. A distinct similarity exists between the symbolic language used by Boole to express logical relationships and the symbology used in ordinary algebra to express mathematical relationships, therefore the term "Boolean algebra." A summary of Boolean facts is presented in table 2-1. As shown, the symbols themselves are the same as those used in modern algebra, but their meanings and usage may have been modified to fit Boolean algebra.

Q2-1. Refer to table 2-1. What symbol indicates the logical sum operation?

Q2-2. Refer to table 2-1. What symbol(s) indicate(s) that all terms are to be treated as a unit?

RULES OF BOOLEAN ALGEBRA

Boolean algebra is a two-value system of logical representation. A variable can have either of two possible values: if $A \neq 0$ then $A = 1$, and if $A \neq 1$ then $A = 0$.

Q2-3. According to Boolean algebra how many values can be assigned to the term A?

The preceding rules, although simple in appearance, may be used to construct a Boolean algebra table to determine all of the following relationships:

Logical Addition (OR Function)	$0 + 0 = 0$ $0 + 1 = 1$ $1 + 0 = 1$ $1 + 1 = 1$
Logical Multiplication (AND Function)	$0 \cdot 0 = 0$ $0 \cdot 1 = 0$ $1 \cdot 0 = 0$ $1 \cdot 1 = 1$
Rules of Complement (NOT Function)	$\overline{0} = 1$ $\overline{1} = 0$

The theorems shown in table 2-2 (foldout at the end of this chapter) include most of the basic rules used in simplifying Boolean expressions and logic circuits. These laws and theorems may be proven by using the preceding 10 basic statements, and by a process known as "proof by perfect induction." The laws and theorems are based on logic and observation. It might be helpful to memorize this table if Boolean algebra will be used on a day-to-day basis.

The Law of Identity

Figure 2-1 shows the logic mechanization, switch mechanization, and truth tables that

Table 2-1.–Boolean connectors and variables

=	The equal sign, just as in conventional mathematics, represents a relationship of equivalence between the expressions so connected.
• or x	The dot or small x indicates the logical product, or conjunction of the terms so connected. The operation is also frequently indicated with no symbol used, i.e., A·B = A x B = AB. Most generally referred to as the AND operation, the terms so related are said to be "ANDed."
+	The plus sign indicates the logical sum operation, a disjunction of the terms so connected. Usually called the OR operation and the terms so connected are said to be "ORed."
———	The vinculum serves a dual purpose. It is at the same time a symbol of grouping and of operation. As a sign of operation it indicates that the term(s) so overlined is/are to be complemented. As a symbol of grouping it collects all terms to be complemented together. Terms so overlined are often said to be NEGATED, the process of taking the complement is then called NEGATION.
( ) [ ] { }	These familiar signs of grouping are used in the customary fashion to indicate that all terms so contained are to be treated as a unit.
A, B, etc.	Various letters are used to represent the variables under consideration, generally starting with A. Since the variables are capable of being in only one of two states the numerals 0 and 1 are the only numbers used in a Boolean expression.

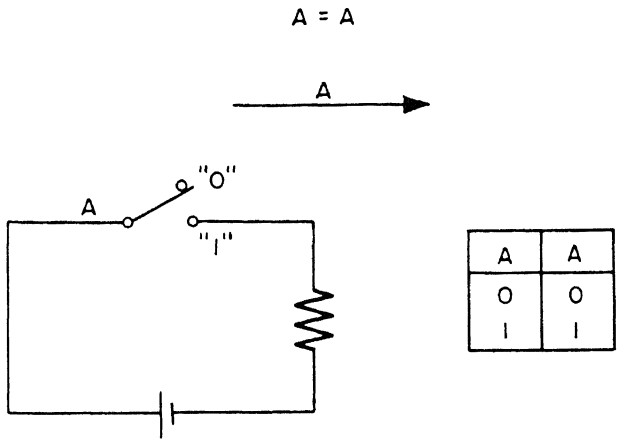


Figure 2-1.–Law of Identity.

apply to the Law of Identity. The Law of Identity states $A = A$, or $\overline{A} = \overline{A}$. This simply means that any expression is equal to itself. For example, if A is true in one part of a Boolean expression, it is true in all parts of that same expression. This is demonstrated by the following example.

$$A + (BC) + AC$$

TRUE TRUE

Commutative Law

Figure 2-2 shows the logic mechanization, switch mechanization, and truth tables that apply to the Commutative Law. The commutative law states $AB = BA$, and $A + B = B + A$. In plain English, the equation states that when logic symbols are ANDed or ORed, the order in which they are written does not affect their value. When given the Boolean expression $ABC = CBA$, and all inputs (ABC) are true, the output is just as true if written as ABC or CBA or BCA. IT IS IMPORTANT TO REALIZE THIS. When simplifying a Boolean expression it is necessary to recognize that one part of the expression is equal to another. Examine the expression BC + ADE + DEA to see

A2-1. The + symbol

A2-2. Parentheses ( ), brackets [ ], and braces { }

A2-3. Two

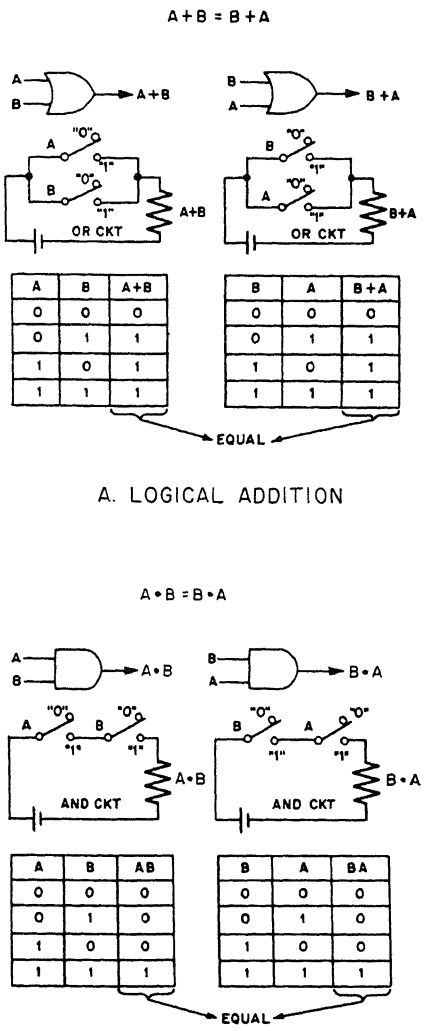
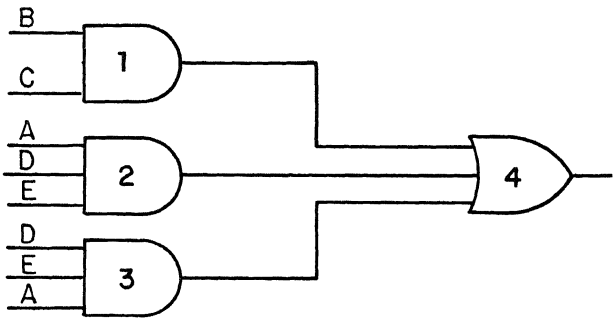


Figure 2-2.—Commutative law.

how this works. The expression can be diagramed as



Note that AND gates 2 and 3 have the same inputs of A, D, and E. The law of identity states that the A of AND gate 2 is equal to the A of AND gate 3. (The same may be said for inputs D and E.) The output will be a true from AND gates 2 and 3 at the same time. Gate 4 is an OR gate requiring only one true input for a true output. Thus, one gate—either 2 or 3—is wasted and can be eliminated. The only time the logic circuit will provide a true output is when B and C are true, or ADE is true. Thus, the expression BC + ADE + DEA can be written as

$$BC + ADE + \cancel{DEA} = BC + ADE$$

Q2-4. To which of the following expressions can the commutative law be applied?

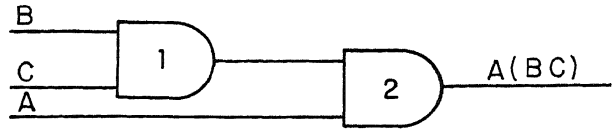
1. (A + D + BC) (CB + D + A)
2. [A(BC + D + $\bar{E}$)] [($\bar{E}$ + BC + D)A]
3. (HI + L) (HL + I)

Q2-5. Simplify the following expressions according to the commutative law.

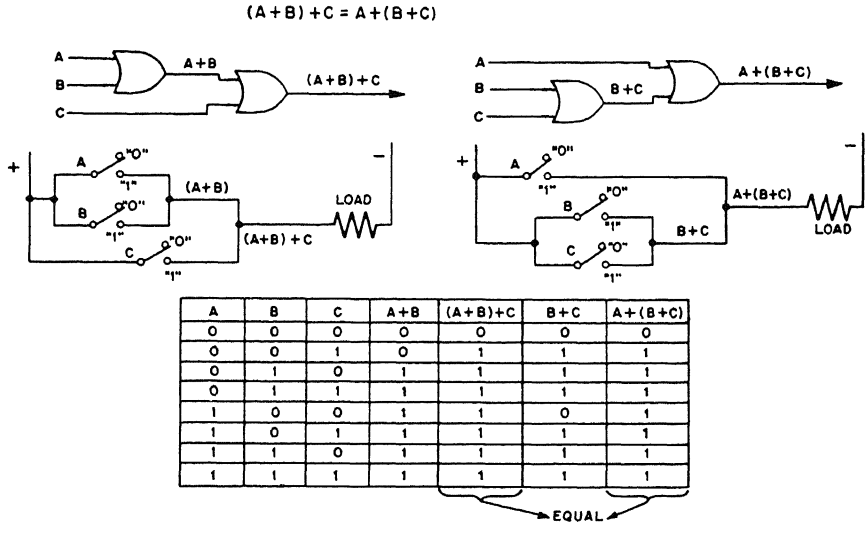
1. $A\bar{B} + \bar{B}A + CDE + \bar{C}DE + E\bar{C}D$
2. $AB + AC + BA$
3. $C\bar{E} + \bar{A} + \overline{CE} + \overline{EC}$
4. $E\bar{B} + AG + \bar{B}E + A\bar{G}$
5. (LMN)(AB)(CDE)(MNL)
6. $F(K + R) + SV + W\overline{X} + VS + \overline{X}W + (R + K)F$

Associative Law

Figure 2-3 shows the logic mechanization, switch mechanization, and truth tables that apply to the Associative Law. The associative law states that A(BC) = ABC, and A + (B + C) = A + B + C. By diagraming the expression A(BC) and A + (B + C), we can prove that the associative law is correct.

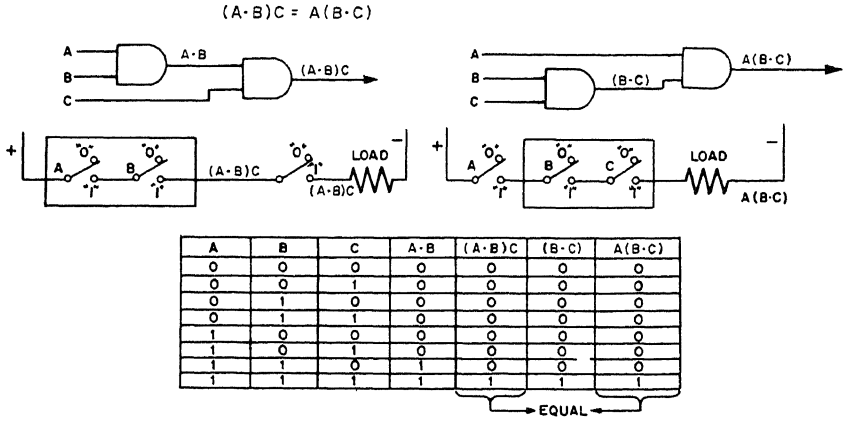


Notice that AND gate 2 in the preceding diagram requires A and the output of AND gate 1 to be true. AND gate 1 requires both B and C



A	B	C	A+B	(A+B)+C	B+C	A+(B+C)
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	1	1	0	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

A. LOGICAL ADDITION



A	B	C	A·B	(A·B)C	(B·C)	A(B·C)
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	0
1	1	1	1	1	1	1

B. LOGICAL MULTIPLICATION

Figure 2-3.–Associative law.

A2-4. 1 and 2

A2-5. 1. $A\bar{B} + CDE + \bar{C}DE$

2. $AB + AC$

3. $\bar{C}\bar{E} + C\bar{E} + \bar{A}$

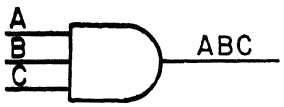
4. $E\bar{B} + AG + A\bar{G}$

5. $(LMN)(AB)(CDE)$

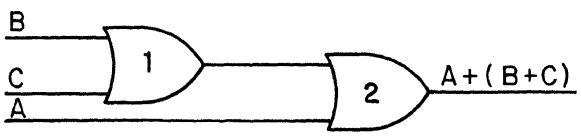
6. $F(K + R) + SV + W\bar{X}$

---

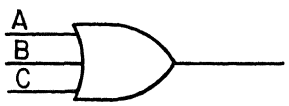
to be true for a true output. The output of AND gate 2 requires that A and B and C all be true for a true output. Thus, A(BC) can be rewritten and rediagramed as ABC, as shown below.



The OR function of A + (B + C) can be treated the same way. First, diagram it as



Anytime B or C is true, OR gate 1 will give a true output which OR gate 2 will pass as a true output. When A is true, OR gate 2 will also give a true output. Thus, OR gate 2 will give a true output when either A or B or C is true. This can be diagramed as



Q2-6. Simplify the following Boolean expressions.

1. ABC(DE)
2. BC + (DE + FG)
3. A(BC) + DC(BE)
4. W + (X + Y) + Z + (V + V)

Idempotent Law

Figure 2-4 shows the logic mechanization, switch mechanization, and truth tables that apply to the Idempotent Law. The idempotent law is one of the easier laws of Boolean to

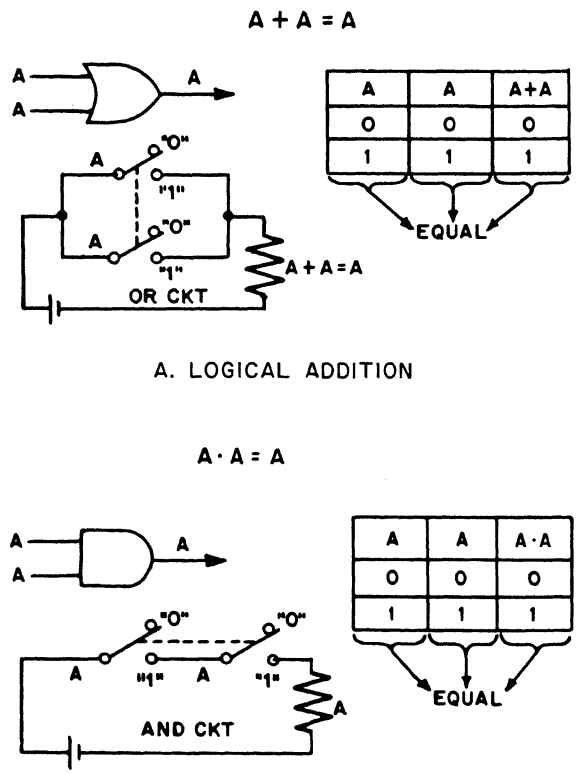
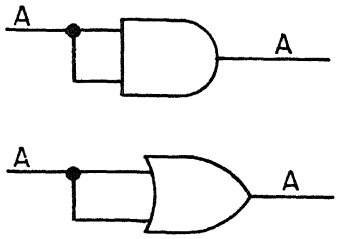
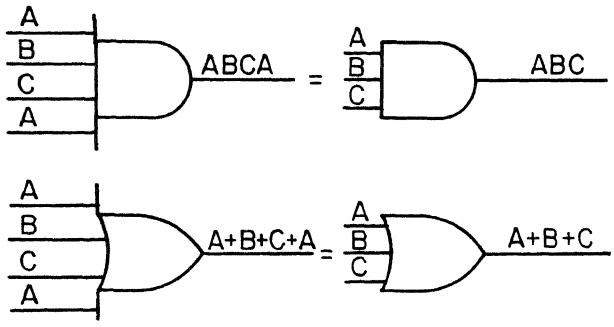


Figure 2-4.—Idempotent law.

understand and use. It can be stated as AA = A, or A + A = A, and diagramed as



Anytime A is true, both inputs will be true. Any term ANDed or ORed with itself will be equal to itself. This means that if A is true and it is ORed with itself, the output will be true. If A is false, then the output will be false. While the expression AA (or A + A) is seldom seen in Boolean, the expression ABCA (or A + B + C + A) may be used, in which case the extra term can be eliminated as in the following illustration.



Q2-7. Simplify the following Boolean expressions. The numbers to the right of the expression indicate which of the first four Boolean laws should be used.

1. **(RS)(RS) + A(BC) + CAB** — **2, 3, 4**
2. **$\overline{X}$YZ + X + (ZY)(YZ) + X + (BC)** — **2, 3, 4**
3. **LMLMNN** — **4**
4. **ABC + A(BC) + D + (E + F)** — **3, 4**

Law of Double Negation

Figure 2-5 shows the logic mechanization, switch mechanization, and truth tables that apply to the Law of Double Negation. The double negation law is written as $\overline{\overline{A}} = A$. This should be perfectly clear. If a NOT expression is brought into an inverter, it is inverted to its opposite state.

Q2-8. Simplify the following examples using the double negation law.

1. $\overline{\overline{\overline{AB}} + \overline{X}}$
2. $\overline{\overline{D}}\,\overline{\overline{E}}$
3. $\overline{(\overline{\overline{R + S}})}\,\overline{T}$

Q2-9. Simplify the following expressions using the double negation law and other laws you have learned.

1. $\overline{\overline{C}} + \overline{DF} + \overline{\overline{FD}} + G + \overline{\overline{C}}$
2. $(\overline{\overline{Q + R} + \overline{S}}) + (R + Q)$
3. $(M\overline{\overline{NP}} + Q)L + \overline{\overline{K}}$
4. $(\overline{\overline{WXY}} + Z)\,(Z + \overline{\overline{XWY}})$

Complementary Law

Figure 2-6 shows the logic mechanization, switch mechanization, and truth tables that apply to the Complementary Law. The complementary law states that $A\overline{A}$ = false or 0, or $A + \overline{A}$ = true or 1.

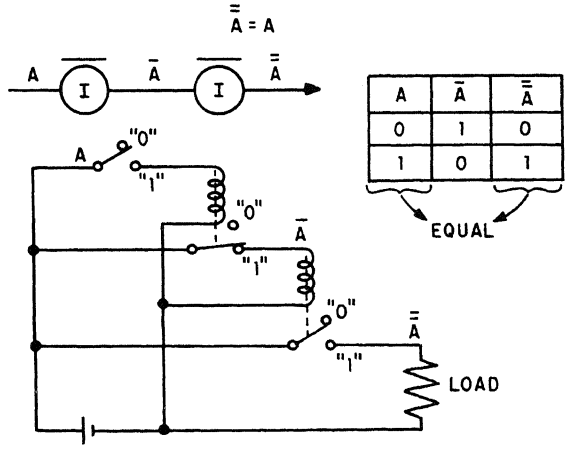


Figure 2-5.—Law of double negation.

A2-6. 1. ABCDE
2. BC + DE + FG
3. ABC + DCBE
4. W + X + Y + Z + V

A2-7. 1. RS + ABC
2. $\overline{X}YZ + X + ZY + BC$
3. LMN
4. ABC + D + E + F

A2-8. 1. $\overline{AB} + X$
2. $\overline{D}\,\overline{E}$
3. $(\overline{R} + S)\overline{T}$

A2-9. 1. $C + \overline{DF} + G$
2. Q + R + S
3. $(MNP + Q)L + \overline{K}$
4. WXY + Z

In the first case, $A\overline{A}$ shows an AND gate with both states of A as inputs. Therefore, one input is true (1) and one is false (0).

AND gates require each input to be true (1) to get a true (1) output. Therefore, the only possible output in this instance is false (0). This is shown in figure 2-6.

The same applies to the OR function of $A + \overline{A}$ = true (1). The OR gate requires that only one input be true (1) to get a true (1) output. Therefore, with inputs of $A + \overline{A}$ the term that represents a true (1) (either A or $\overline{A}$) will cause the OR gate to output a true (1). This is shown in figure 2-6.

Q2-10. Some of the following logic gates have true outputs and some have false outputs. Indicate which outputs are true by marking each with a 1, and those that are false mark with a 0.

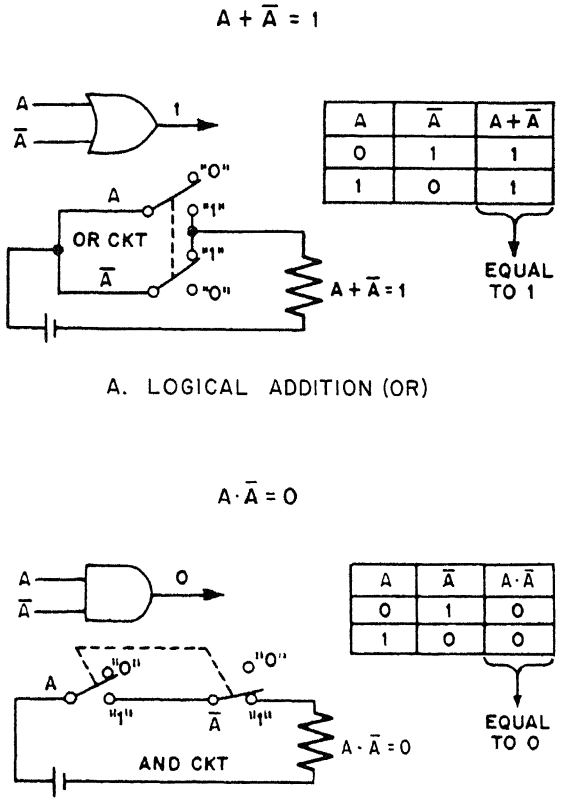
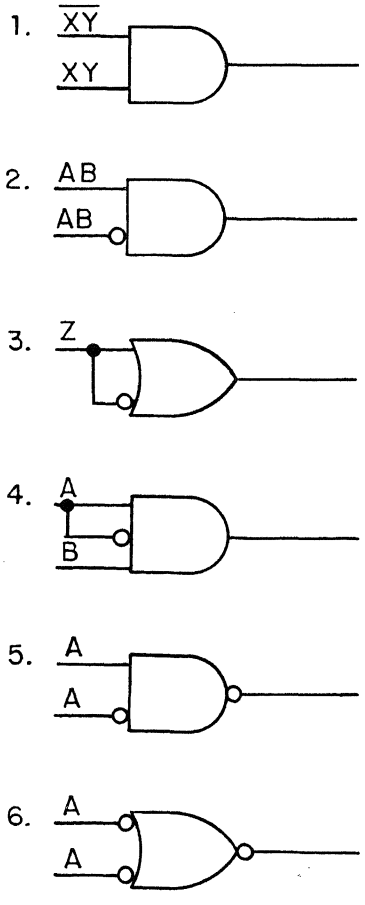


Figure 2-6.—Complementary law.



Now that the first six laws of Boolean have been covered, they will be used to simplify the following expressions.

Example:

$$(A + B)(A + B)\overline{(A + B)}$$

In this example, apply the law of identity first.

$$(A + B) = (A + B)$$

Next, apply the idempotent law.

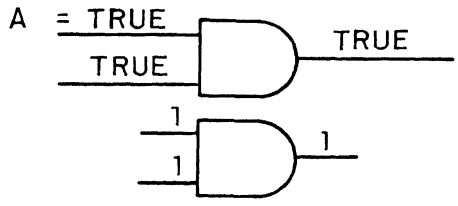
$$(A + B)(A + B) = (A + B)\cancel{(A + B)} = (A + B)$$

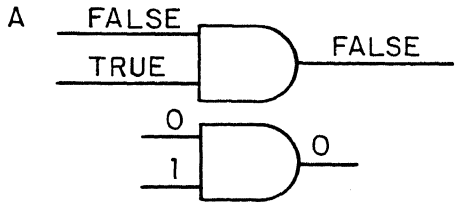
This leaves $(A + B)\overline{(A + B)}$. Since $\overline{(A + B)}$ is the complement of $(A + B)$, the complementary law can be applied. Therefore,

$$(A + B)\overline{(A + B)} = 0$$

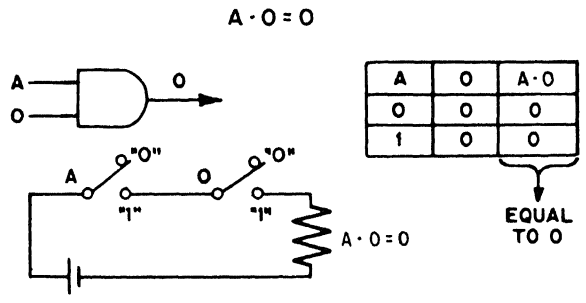
Law of Intersection

Figure 2-7 shows the logic mechanization, switch mechanization, and truth tables that apply to the Law of Intersection. The law of intersection states that $A \cdot \text{true}\ (1) = A$, or $A \cdot \text{false}\ (0) = \text{false}\ (0)$. The truth of this law is obvious. In the case of $A \cdot \text{true}\ (1) = A$, if one input to an AND gate is always true (1), then the output will depend on the state of the other input (in this case A). If A is true (1), the AND gate will have two true (1) inputs and the output will be true (1). If A is false (0), the AND gate will have one false (0) input (in this case A) and one true (1) input; therefore, the output will be false (0). In both cases, the output is the same state as A. This may be shown by the following diagram.





In the case of $A \cdot \text{false}\ (0) = \text{false}\ (0)$, if one input to an AND gate is always false (0), the output will always be false (0).



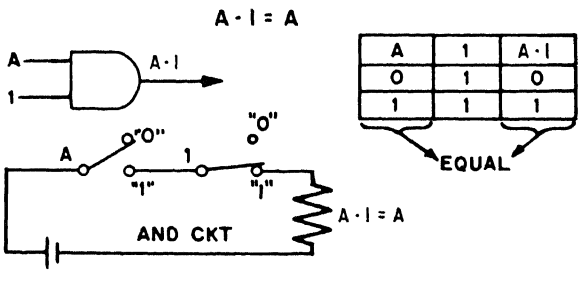


Figure 2-7.—Law of intersection.

Law of Union

Figure 2-8 shows the logic mechanization, switch mechanization, and truth tables that apply to the Law of Union. The law of union states that $A + \text{true}\ (1) = \text{true}\ (1)$, or $A + \text{false}\ (0) = A$. In the first statement, $A + \text{true}\ (1) = \text{true}$, the OR gate has one input labeled true (1). Its output, therefore, will be a true (1) regardless of the state of A. The statement $A + \text{false}\ (0) = A$ is just the opposite. One input is labeled false (0) and the only way to get a true (1) output from the OR gate is if A

A2-10. 1. 0

2. 0

3. 1

4. 0

5. 1

6. 0

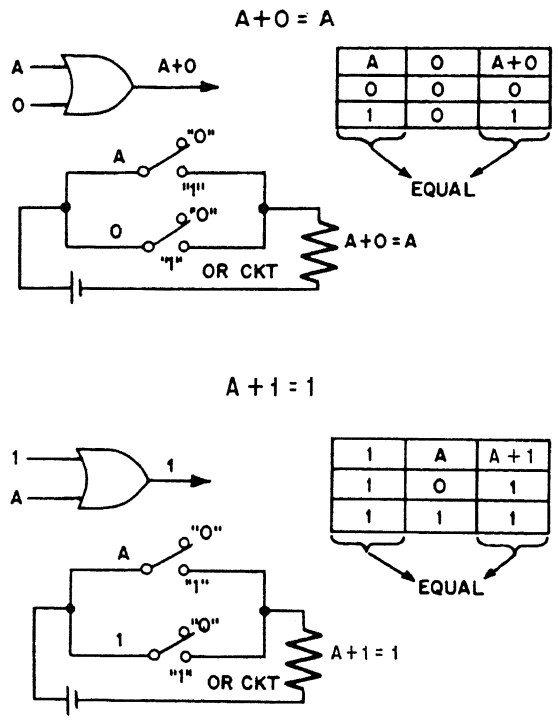


Figure 2-8.—Law of union.

is true (1). Therefore, the state of A will determine the output state of the OR gate. These conditions are shown in figure 2-8.

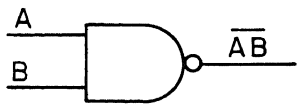
As can be seen, when A is ORed with a true (1), the output will always be true (1), regardless of the state of A. Likewise, when A is ORed with a false (0), the output will depend on the state of A.

Q2-11. Use the laws of intersection and union to simplify the following expressions.

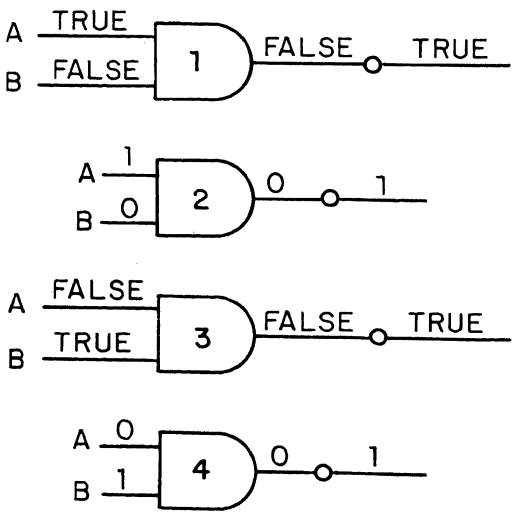
1. true · $\overline{B}$
2. 1 + A
3. 0 · A
4. $A\overline{B}$ · 0
5. $(\overline{AC})$ · true

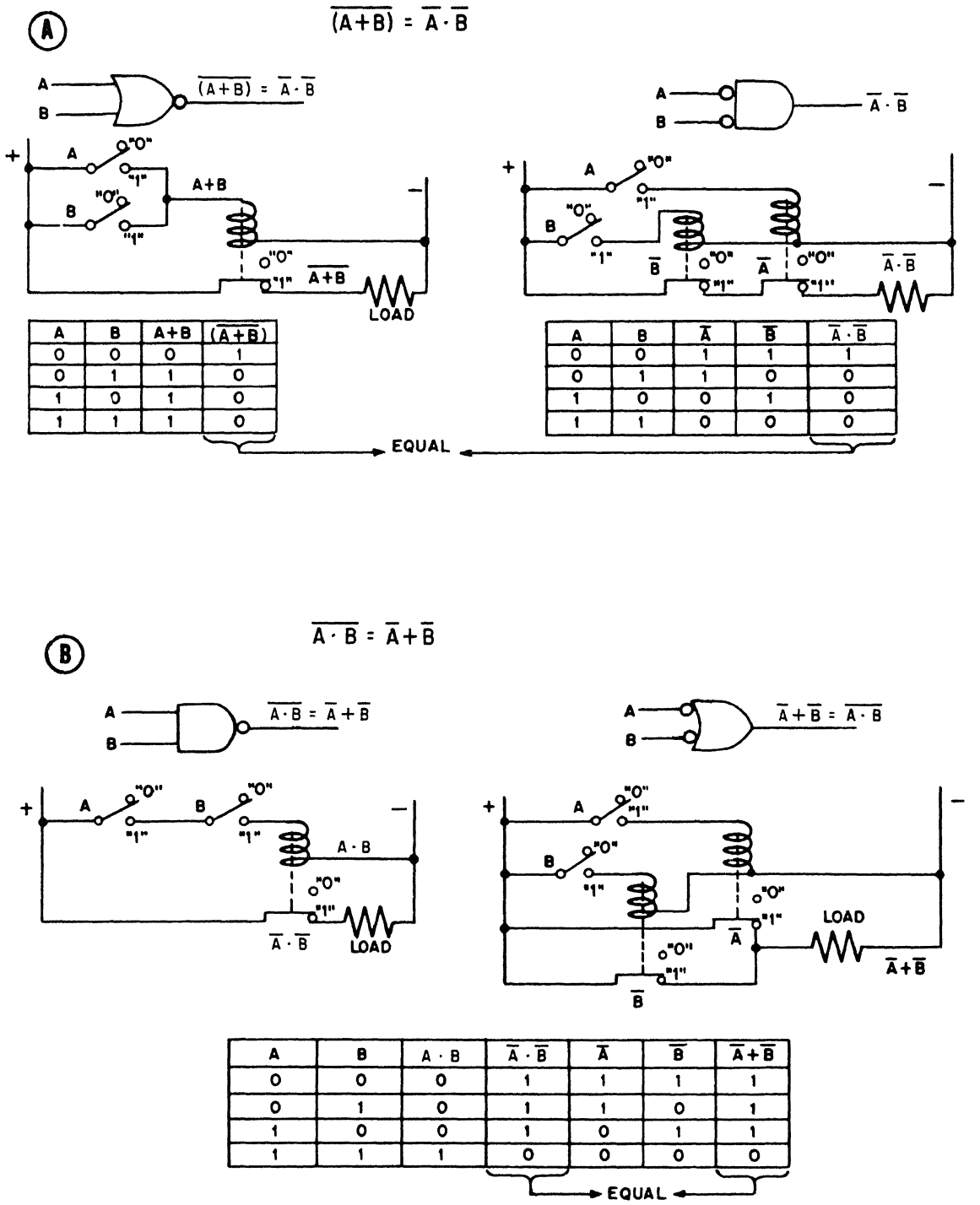
DeMorgan's Theorem

Figure 2-9 shows the logic mechanization, switch mechanization, and truth tables that apply to DeMorgan's Theorem. DeMorgan's theorem is concerned with NAND and NOR logic gates. The first part of the law states that $\overline{AB} = \overline{A} + \overline{B}$. The solid vinculum indicates the presence of the NAND gate, as shown in the following diagram.



Notice, in the diagram of NAND gates that the only time a true (1) is present at the output from a NAND gate is if one of the inputs is false (0).





A	B	A+B	$\overline{(A+B)}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

A	B	$\overline{A}$	$\overline{B}$	$\overline{A} \cdot \overline{B}$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

A	B	A · B	$\overline{A \cdot B}$	$\overline{A}$	$\overline{B}$	$\overline{A} + \overline{B}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

Figure 2-9.—DeMorgan's theorem.

A2-11. 1. $\overline{B}$

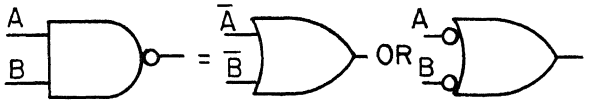
2. 1

3. 0

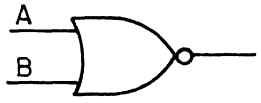
4. 0

5. $\overline{AC}$

In each of the preceding NAND gates, it is the false, or 0, input that causes the true, or 1, output. In NAND gates 1 and 2 the output is a true (1) when B is false (0). (Remember, $\overline{B}$ is B complemented.) In NAND gates 3 and 4 the output is true (1) when A is false (0). Therefore, in the NAND gates pictured, anytime that $\overline{A}$ or $\overline{B}$ is true (1), the output will be a true (1), or, as the law states, $\overline{AB} = \overline{A} + \overline{B}$.



The second half of DeMorgan's theorem is stated as $\overline{A + B} = \overline{A}\overline{B}$. The term $\overline{A + B}$ is obviously the output from a NOR circuit and can be diagramed as



The only time the output is true from a NOR circuit is when both inputs are false. When A and B are both false, $\overline{A}$ and $\overline{B}$ are both true. It can be stated that $\overline{A + B} = \overline{A}\overline{B}$, and that both of these expressions will equal a true at the same time.

DeMorgan's theorem is one of the most useful Boolean tools presented. It should be mastered. It can be used to split or join vincula and thus greatly simplify most Boolean expressions. The expression $RS + TV + \overline{R} + \overline{S}$ can be simplified to its basic terms by the use of DeMorgan's theorem.

Step 1: Apply DeMorgan's theorem.

$$RS + TV + \overline{RS}$$

Step 2: Apply the commutative law.

$$RS + \overline{RS} + TV$$

Step 3: Apply the complementary law.

$$RS + \overline{RS} = 1$$

$$1 + TV$$

Step 4: Apply the law of union.

$$1 + TV = 1$$

The answer is 1 or true. Now, apply DeMorgan's theorem to the following problems.

Q2-12. 1. $\overline{V + A + L}$

2. $\overline{A} + \overline{B} + \overline{C} + \overline{D}$

3. $\overline{WXYZ}$

4. $\overline{ABC} + D$

So far, most of the problems have had a single vinculum over them. The following section explains the process of simplifying expressions with more than one vinculum over them. For example,

$$\overline{\overline{B} + CD}$$

First, split the long vinculum and change the signs (using the single-step operation shown above) until the vinculum is completely split and all signs have been changed.

$$\overline{\overline{B}}(\overline{C} + \overline{D})$$

Then, apply the double negation law.

$$B(\overline{C} + \overline{D})$$

Q2-13. Simplify the following expressions.

1. $\overline{(A + B)(\overline{L + M})}$

2. $\overline{WX\overline{Y}} + \overline{A\overline{B}}$

Use DeMorgan's theorem not only to combine and split vincula, but, when used with the double negation law, to add vincula to expressions. By using both methods, it is very easy to simplify an expression.

Following is a summary of the principles and applications of DeMorgan's theorem.

1. DeMorgan's theorem is used to split or join vincula.

$$\overline{A\overline{B}} = \overline{A} + \overline{B}$$

$$\overline{A + B} = \overline{A}\,\overline{B}$$

2. When a vinculum is split or when several vincula are joined to form one long vinculum every sign over which the splitting or joining takes place changes from OR to AND, or from AND to OR.

3. When changing signs, preserve the original grouping.

$$\overline{A + BC} = \overline{A}(\overline{B} + \overline{C})$$

4. An expression is in its simplest form only if no letter has more than one vinculum over it.

5. There are two ways to simplify an expression by applying DeMorgan's theorem.

a. Join vincula to form the complement of a term or expression.

$AB + C + \overline{A} + \overline{B}$
$AB + C + \overline{A\overline{B}}$
$1 + C$
1

b. Split all vincula, then apply the other laws for further simplification.

$\overline{\overline{(AB + \overline{C})DC}}$

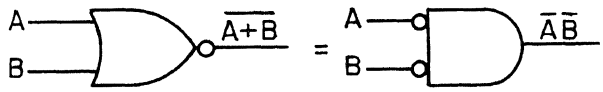
$\overline{\overline{AB + \overline{C} + \overline{DC}}}$

$AB + \overline{C} + \overline{DC}$

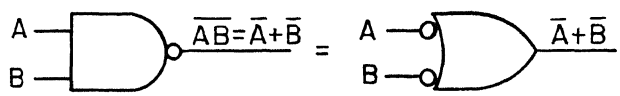
$AB + \overline{C} + \overline{D} + \overline{C}$

$AB + \overline{C} + \overline{D}$

6. NOR gate = AND gate with inverted inputs.



NAND gate = OR gate with inverted inputs.



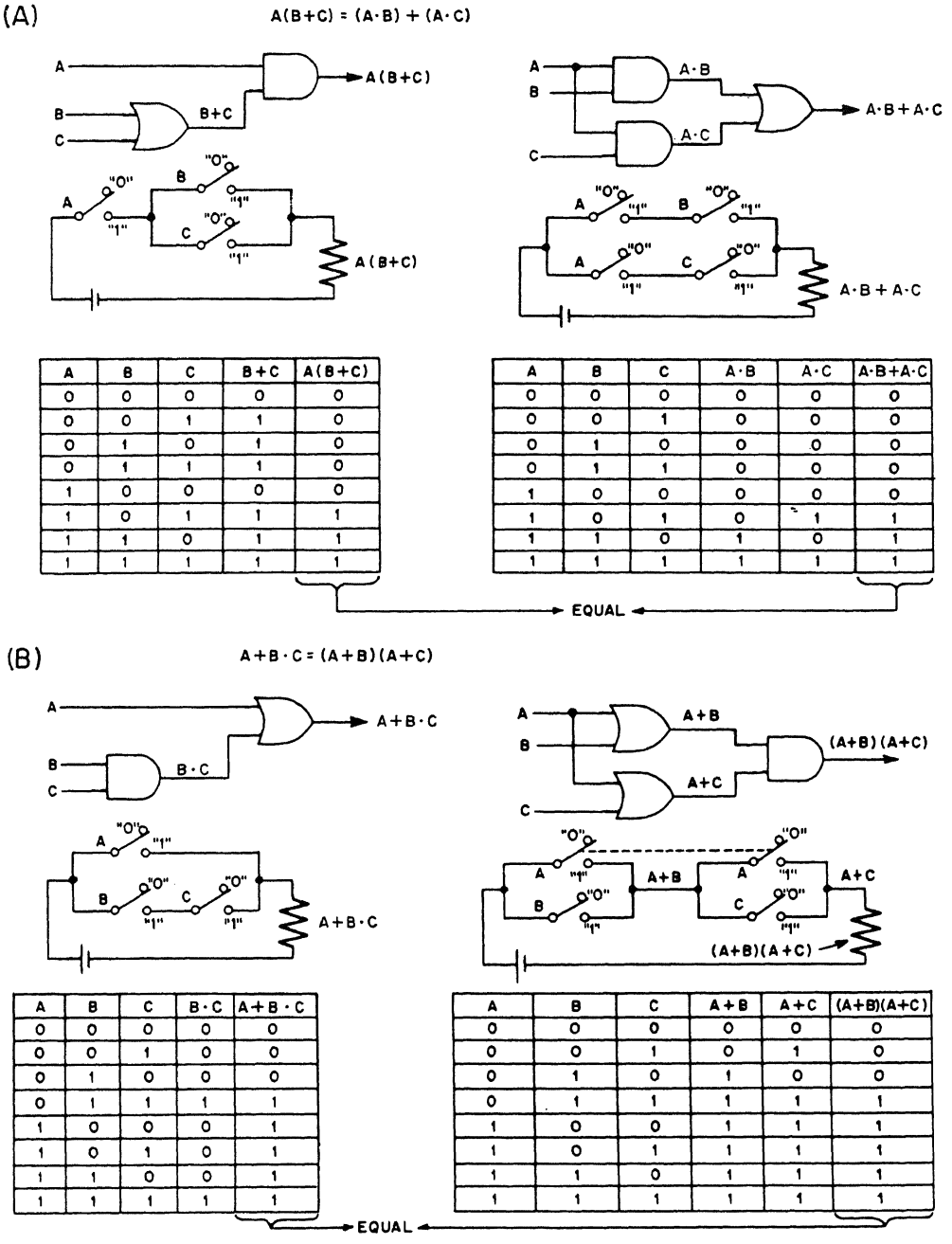
Distributive Law

Figure 2-10 shows the logic mechanization, switch mechanization, and truth tables that apply to the Distributive Law. The last three laws in table 2-2 are used mainly to manipulate Boolean expressions so that the other laws may be applied to simplify the expression.

The distributive law states that $A(B + C) = AB + AC$. For those familiar with normal algebra, this is nothing more than multiplication of terms. However, since this is a course in Boolean algebra, all laws must be provable using logic gates.

A2-12.
1. $\overline{V}\overline{A}\overline{L}$
2. $\overline{ABCD}$
3. $\overline{W}+\overline{X}+\overline{Y}+\overline{Z}$
4. $\overline{A}+\overline{B}+\overline{C}+D$

A2-13.
1. $\overline{A}\overline{B}+L+M$
2. $(\overline{W}+\overline{X}+Y)+(\overline{A}+B)$



A	B	C	B+C	A(B+C)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

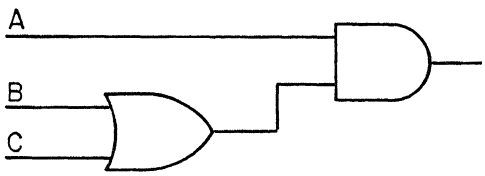
A	B	C	A·B	A·C	A·B+A·C
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

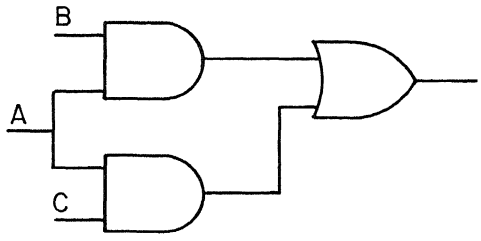
A	B	C	B·C	A+B·C
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

A	B	C	A+B	A+C	(A+B)(A+C)
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

Figure 2-10.—Distributive law.

To get a 1 or true output from A(B + C), A must be true and either B or C must be true. There are two possible combinations of logic that will provide a true output at the same time as A(B + C). These are, AB + AC, as shown below.





While this may seem to violate the purpose of simplification using Boolean algebra, it really doesn't. As stated previously, the main purpose of the distributive law is manipulation. For example, the expression W(X + Y) + WY can not be simplified without manipulation. Apply the distributive law to get WX + WY + WY. Then, by applying the idempotent law the expression is WX + WY.

Q2-14. Apply the distributive law to the following expressions.

1. D(E + F + G)
2. A(A + B + D)
3. V(W + Y + XZ)
4. JK(L + MN)
5. $\overline{H}J + JK\overline{L} + GJM$
6. LMNP + LQR

Q2-15. Use the distributive law in conjunction with the laws already covered to simplify the following expressions.

1. A(WX + AB + A) + AWX
2. ZW(ABWZ) + A + (A + B)
3. $X(\overline{X}B) + A(\overline{\overline{A}C + B})$
4. $(TV)(T + W)(U + V)(V + W)(\overline{TV}) + Y$
5. (BC + GB + JG)BJ
6. $AC + AD + A\overline{C} + (B + D)(\overline{B}\overline{D})$
7. $[MN(X + Y + Z) + YNM]\,[\overline{M} + \overline{N} + \overline{Z}]$

Law of Absorption

Figure 2-11 shows the logic mechanization, switch mechanization, and truth tables that apply to the Law of Absorption. The next law on the foldout is the law of absorption. It states that A(A + B) = A, or A + AB = A.

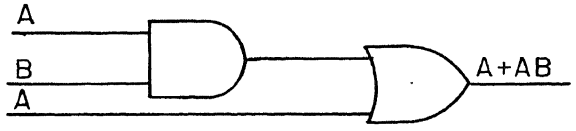
The law of absorption is easy enough to prove. First, apply the distributive law to A(A + B).

$$AA + AB$$

Then, the idempotent law

$$AA + AB = A + AB$$

Then, draw out the logic diagram.



Anytime that A is true, the output of the OR gate will be a true. Anytime A is false, the output from either the AND or the OR gate will be a false. Therefore, anytime A is equal to 1 from either the AND or OR gate, the output will be a 1, regardless of the state of B.

Apply the distributive law to the expression A + AB = A(1+B). How was the expression 1 + B derived? This is an application

A2-14.
1. DE + DF + DG
2. AA + AB + AD
3. VW + VY + VXZ
4. JKL + JKMN
5. $J(\bar{H} + K\bar{L} + GM)$
6. L(MNP + QR)

A2-15.
1. AWX + AB + A = A
2. ZWAB + A + B
3. $A + A\bar{C}\bar{B}$
4. Y
5. BJC + BJG
6. A
7. $MN\bar{Z}X + MN\bar{Z}Y$

of the law of intersection which states that A = A · 1. Therefore, the original expression could have been written as A(1) + AB, in which case it would mean that both 1 and B were ANDed together with A. If A(1 + B) is correct, the law of union, which states that A + 1 = 1, is then applied; therefore, the expression 1 + B is equal to 1 and the result is that A · 1 = A.

Taken in steps, the preceding example would look like this:

Step 1: A + AB

Step 2: Apply the distributive and intersection laws.

A(1 + B)

Step 3: Apply the law of union.

A · 1

Step 4: Apply the law of intersection.

A

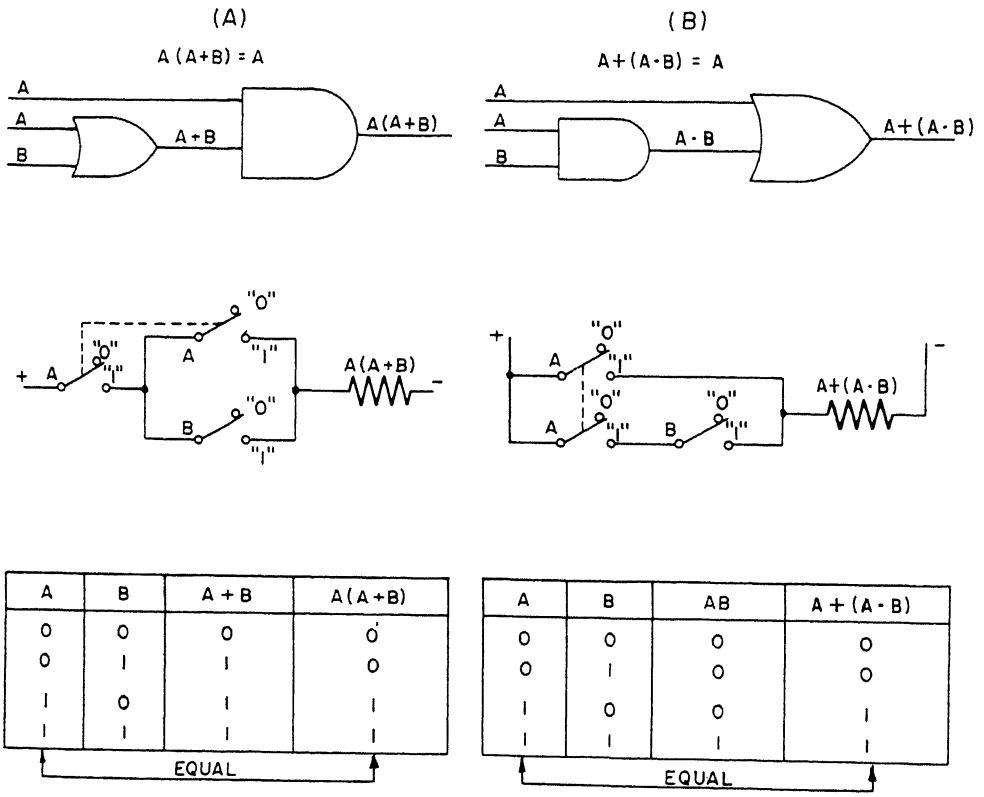


Figure 2-11.—Law of absorption.

As another exercise, simplify VW + W + WX according to the law of absorption.

1. First, factor out the common term according to the distributive law.

$$W(V + 1 + X)$$

2. Check to see if the law of union applies. If it does, rewrite the problem as:

$$W(V + 1 + X) = W(1)$$

3. Apply the law of intersection.

$$W \cdot 1 = W$$

The law of absorption can also be used to simplify a different form of expression. For example, if given A(B + C + A), first apply the distributive law.

$$A(B + C + A) = AB + AC + AA$$

Next, apply the idempotent law.

$$AB + AC + AA = AB + AC + A$$

Reapply the distributive law to factor out the common term.

$$AB + AC + A = A(B + C + 1)$$

Apply the law of union.

$$A(B + C + 1) = A(1)$$

Then, apply the law of intersection.

$$A(1) = A$$

Q2-16. Simplify the following expressions.

1. **R(S + T + R)**
2. **(AB + DE + V)DE**
3. **(XY + WZ + X)X**
4. **K + KL + KM**
5. **VW + W + WX**

At this point it should be noted that most laws of Boolean can be applied both backward and forward. This means that the equal signs can be replaced with double arrows. For example:

$$A + (B + C) = A + B + C$$

or

$$A + (B + C) \leftrightarrows A + B + C$$

To simplify, look at all expressions from both directions of the laws given in table 2-2.

Q2-17. With this fact in mind, use laws that have already been covered to simplify the following expressions.

1. **ST + VW + RST**
2. **TUV + XY + Y**
3. **F(E + F + G)**
4. **(PQ + R + ST)TS**
5. **ABC + CB**
6. $\overline{\overline{D}\overline{D}E}$
7. $Y(W + X + \overline{\overline{Y} + \overline{Z}})Z$
8. **JKL + J**
9. **(BE + C + F)C**
10. $MNP + QR + \overline{\overline{M} + \overline{N}}$

Law of Common Identities

The last law used for Boolean simplification is the Law of Common Identities. It is nothing more than an application of the distributive, complementary, union, and intersection laws.

A2-16.

1. R
2. DE
3. X
4. K
5. W

A2-17.

1. ST + VW
2. TUV + Y
3. F
4. ST
5. BC
6. D
7. YZ
8. J
9. C
10. MN + QR

The law states $A(\bar{A} + B) = AB$, or $A + \bar{A}B = A + B$. Take the first part, $A(\bar{A} + B) = AB$, and apply the following laws.

Step 1: The distributive law

$$A(\bar{A} + B) = \bar{A}A + AB$$

Step 2: The complementary law

$$A\bar{A} + AB = 0 + AB$$

Step 3: The law of union

$$0 + AB = AB$$

This is the proof for the law of common identities of

$$A(\bar{A} + B) = AB$$

The common identities law can be proven using the distributive law, $(A + BC) = (A + B)(A + C)$. Look at the common identities law.

$$A + \bar{A}B = A + B$$

Step 1: Apply the distributive law.

$$A + \bar{A}B = (A + \bar{A})(A + B)$$

Step 2: Apply the complementary law.

$$(A + \bar{A})(A + B) = 1(A + B)$$

Step 3: Apply the law of intersection.

$$1(A + B) = A + B$$

Expressions like $A(\bar{A} + B)$ and $A + AB$ occur frequently in Boolean algebra. Learning to recognize that $A(\bar{A} + B) = AB$ and $A + AB = A + B$, will help simplify these expressions more rapidly.

THE VEITCH DIAGRAM

Now that the preceding problems using the laws of Boolean simplification have been covered, two drawbacks may become evident. The first is that some Boolean expressions require many time-consuming steps to simplify. Secondly, unless one becomes very familiar with all the possible ways that the laws of Boolean can be applied, it is difficult to realize just when a Boolean expression is in its simplest form.

Fortunately there is another method that can be used for Boolean simplification–the VEITCH DIAGRAM. A Veitch diagram provides a very quick and easy way for finding the simplest logic equation needed to express a function. But, before going into the construction of Veitch diagrams, it is first necessary to learn how to set up a Boolean expression for insertion into a Veitch diagram.

Every character in a Boolean expression has two states that are complements to each other. These complementary states are called variables. The variable A for example would have A and $\bar{A}$ states. If three variables are ANDed together, such as ABC, there are eight possible combinations. These are:

ABC	$\bar{A}\bar{B}C$
$AB\bar{C}$	$\bar{A}BC$
$A\bar{B}\bar{C}$	$\bar{A}B\bar{C}$
$\bar{A}\bar{B}\bar{C}$	$A\bar{B}C$

These possible combinations are called MINTERMS. To be classified as a minterm, an expression MUST be an AND product.

Because each variable has two possible states, the number of combinations can be expressed as the number of possible states (two) raised to a power dictated by the number of variables.

For example, an expression with two variables (A and B) will have four minterms or 2^2.

A · B

two states — two states

A and $\overline{A}$ — B and $\overline{B}$

$$2 \times 2 = 4$$

If an expression contains three variables it will have eight minterms:

$$ABC$$

$$2 \times 2 \times 2 = 8$$

If it contains four variables it will have sixteen minterms:

$$ABCD$$

$$2 \times 2 \times 2 \times 2 = 16$$

Before expressions can be plotted on a Veitch diagram they must be placed in minterm form. This means that all logical operations have to be performed. For example, A(B + C) is not a minterm because the AND operation has not been completed. AB + AC is the minterm form. Any expression can be converted to minterm by using the Boolean laws already covered.

Q2-18. Convert the following expressions to minterm form.

1. $\overline{T + V} + RS$
2. $A + B + \overline{CD} + \overline{ABC}$
3. $\overline{W}\overline{X}YZ + X(W + \overline{Y})$
4. $A(B + C + \overline{D}) + E + AB\overline{C} + \overline{\overline{D}CE}$
5. $ABC + \overline{WX}XW + MNOM$

Once an expression is in minterm form, it is ready to be set up on the Veitch diagram. The Veitch diagram is constructed in the following manner.

Step 1: To provide a box for every possible minterm, determine the number of variables in the equation. Raise 2 to the power that corresponds to the number of variables; that is, $2^2 = 4$, $2^3 = 8$ and $2^4 = 16$, for two-, three-, and four-variable equations, respectively. For a three-variable equation such as $ABC + AB\overline{C} + C\overline{B}A + B\overline{A}C$, eight boxes are necessary.

Step 2: Label each box. The Veitch diagram must include all possible minterms.

	A		$\overline{A}$	
B	$AB\overline{C}$ 1	ABC 2	$\overline{A}BC$ 5	$\overline{A}B\overline{C}$ 6
$\overline{B}$	$A\overline{B}\overline{C}$ 3	$A\overline{B}C$ 4	$\overline{A}\overline{B}C$ 7	$\overline{A}\overline{B}\overline{C}$ 8
	$\overline{C}$	C		$\overline{C}$

Each variable must include half of the boxes available. For example, A in the above Veitch diagram includes boxes 1, 2, 3, and 4, while $\overline{B}$ includes boxes 3, 4, 7, and 8. Each variable must overlap both conditions of the other variables. For example, if ABC is plotted on a Veitch diagram, A must overlap B, $\overline{B}$, C, and $\overline{C}$ (as shown in the previous diagram).

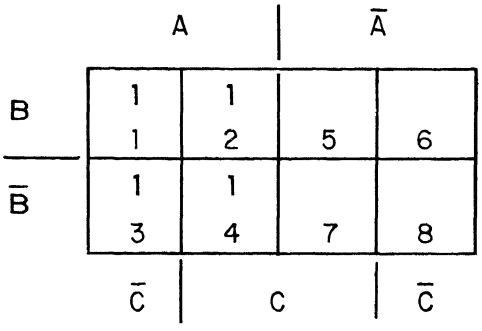
Step 3: To plot an expression on a Veitch diagram, simply mark a 1 in each box which is indicated in the expression. For example, plot the expression A + B + AC.

	A		$\overline{A}$	
B	1	2	5	6
$\overline{B}$	3	4	7	8
	$\overline{C}$	C		$\overline{C}$

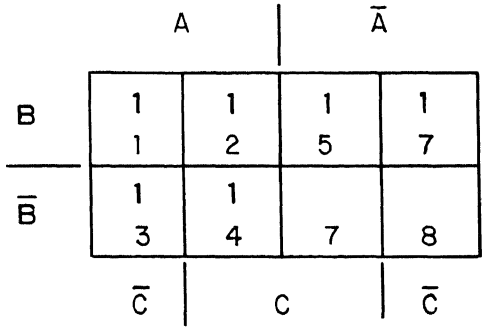
A2-18.

1. $\overline{T} \cdot \overline{V} + RS$
2. $A + B + \overline{C} + \overline{D} + \overline{A} + \overline{B} + C = \overline{D}$
3. $\overline{W}\overline{X}YZ + WX + X\overline{Y}$
4. $AB + AC + A\overline{D} + E + AB\overline{C} + \overline{D}C\overline{E}$
5. $ABC + \overline{WX}XW + MNOM = ABC + MNO$

Place a 1 in all boxes that contain the term A (boxes 1, 2, 3, and 4).



Next, place a 1 in all boxes that contain the term B (boxes 1, 2, 5, and 6). Since boxes 1 and 2 already have 1's, it is not necessary to put more than one 1 in each box.

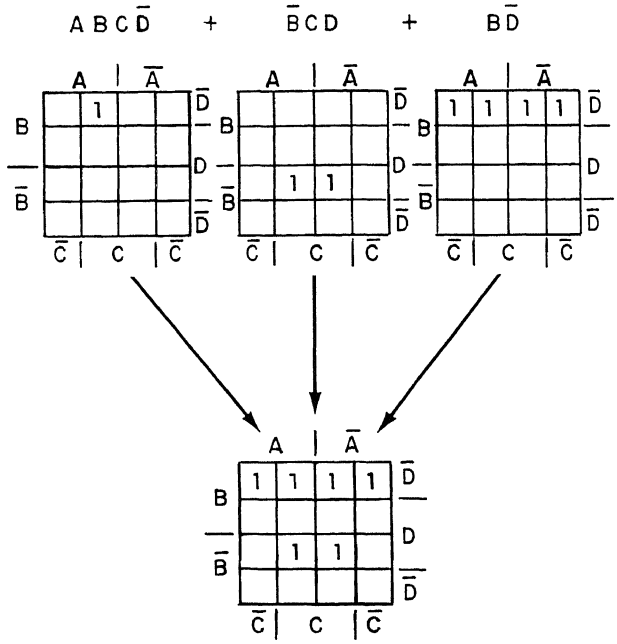


To complete the Veitch diagram, place a 1 in all boxes that contain the term AC (boxes 2 and 4).

Q2-19. Plot the following expressions.

1. $\overline{A}\overline{B}\overline{C} + A\overline{B}C + A$
2. $BC + A$
3. $A + \overline{A}B + C$

The 16- and 32-square Veitch diagrams are plotted in the same manner as the 8-square. The following illustrations are examples of 16-square Veitch diagrams for four variables.



The previous information covered the plotting of minterm form expressions on the various Veitch diagrams. The next step is to extract the simplest expression from the plot. Start with the three-variable expression $\overline{A}BC + \overline{A}B\overline{C} + \overline{A}\overline{B}$, plotted as shown in figure 2-12. Next, extract the simplified expression by looking for patterns in the order of preference.

- Four squares joined together will describe a one-variable term.
- Two squares joined together will describe a two-variable term.
- One square that stands alone will describe a three-variable term.

Join all the adjacent squares into as large a grouping as possible that corresponds to a power of 2-four squares, two squares, or one square. Write the expression that is common to all squares of each group.

In the example, only $\overline{A}$ is common to all the squares grouped together. Therefore, $\overline{A}$ is the

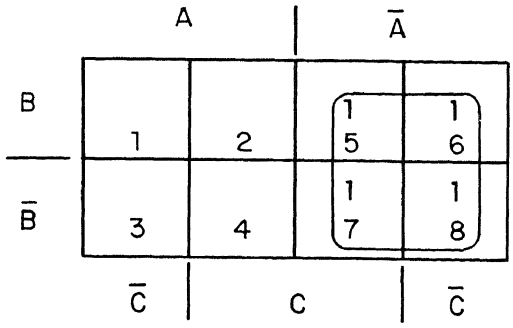
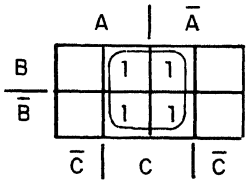


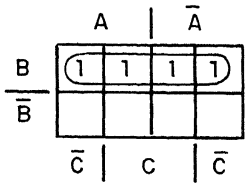
Figure 2-12.–Order of preference for a three-variable expression.

simplified expression for $\overline{A}BC + \overline{A}B\overline{C} + \overline{A}\overline{B}$. (Note that $\overline{A}\overline{B}$ is plotted in both squares 7 and 8.)

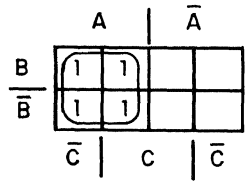
The following Veitch diagrams show some possible groupings of four squares.



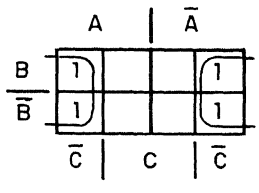
$= C$



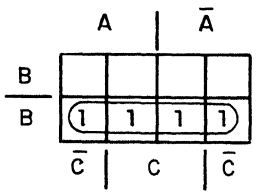
$= B$



$= A$

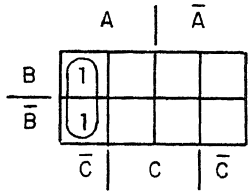


$= \overline{C}$

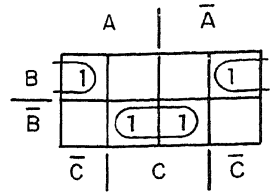


$= \overline{B}$

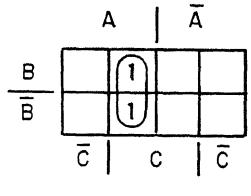
If an expression can not be simplified to a single-variable expression, try to simplify the expression to a two-variable term. This is done by linking the squares into groups of two. Some of the possible patterns for groups of two are:



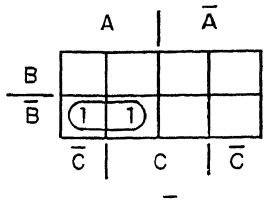
$A\overline{C}$



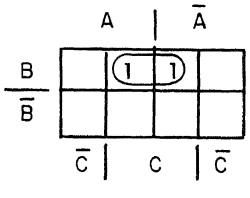
$\overline{B}C + B\overline{C}$



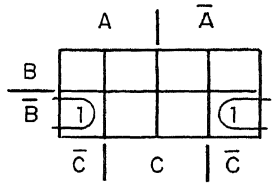
AC



$A\overline{B}$



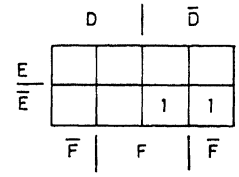
BC



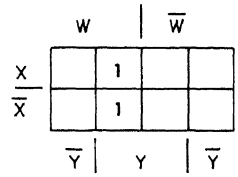
$\overline{B}\,\overline{C}$

Q2-20. Describe the following plots as simply as possible.

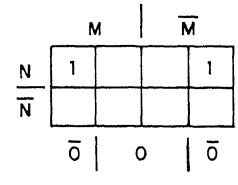
1.



2.



3.



A2-19.

1.

B \ A	A	$\bar{A}$	$\bar{A}$
B	1	1	
$\bar{B}$	1	1	1
	$\bar{C}$	C	$\bar{C}$

2.

B \ A	A	$\bar{A}$	$\bar{A}$
B	1	1	1
$\bar{B}$	1	1	
	$\bar{C}$	C	$\bar{C}$

3.

B \ A	A	$\bar{A}$	$\bar{A}$
B	1	1	1
$\bar{B}$	1	1	1
	$\bar{C}$	C	$\bar{C}$

A2-20. 1. $\bar{D}\bar{E}$

2. WY

3. $N\bar{O}$

So far, all the minterms used have been described by groups of squares that are relatively easy to describe; but, how should the following be described?

B \ A	A	$\bar{A}$	$\bar{A}$
B	1		1
$\bar{B}$	1	1	1
	$\bar{C}$	C	$\bar{C}$

First, try to group all the squares into groups of four, if possible.

B \ A	A	$\bar{A}$	$\bar{A}$
B	1		1
$\bar{B}$	1	1	1
	$\bar{C}$	C	$\bar{C}$

B \ A	A	$\bar{A}$	$\bar{A}$
B	1		1
$\bar{B}$	1	1	1
	$\bar{C}$	C	$\bar{C}$

Notice that some squares are used more than once. This is permissible. The simplified expression is $\bar{B} + \bar{C}$.

Q2-21. Simplify the following expressions.

1. $JK\bar{L} + JKL + J\bar{K}\bar{L} + \bar{J}KL + \bar{J}K\bar{L} + \bar{J}\bar{K}\bar{L}$
2. $XY\bar{Z} + X\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}YZ + \bar{X}Y\bar{Z} + \bar{X}\bar{Y}\bar{Z}$
3. $MN\bar{P} + M\bar{N}\bar{P} + M\bar{N}P + \bar{M}\bar{N}\bar{P}$
4. $\bar{B}CD + B\bar{C}\bar{D} + \bar{B}C + BC\bar{D} + B\bar{C}D + BC$

SUMMARY OF THREE-VARIABLE VEITCH DIAGRAMS

1. An expression can often be simplified most efficiently by plotting it on a Veitch diagram, then extracting the simplified expression from the plot.

2. To plot an expression on a Veitch diagram, convert it to minterm form.

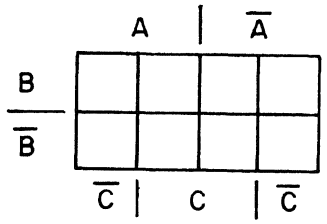
3. To convert an expression to minterm form:

a. Split or remove vincula ($\overline{A + B} = \overline{A}\overline{B}$; $\overline{\overline{A}} = A$)
b. Remove the parentheses (distributive law)
c. Simplify within the term ($ABCA = ABC$; $ABC\overline{A} = 0$)

4. Determine the number of squares needed in the Veitch diagram by using the number of variables as a power of 2.

3 variables–2^3 = 8 squares
4 variables–2^4 = 16 squares
5 variables–2^5 = 32 squares

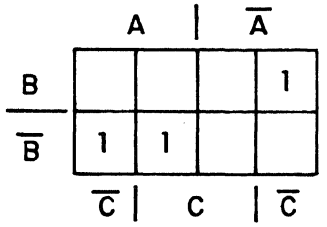
5. A Veitch diagram for variables A, B, and C is labeled:



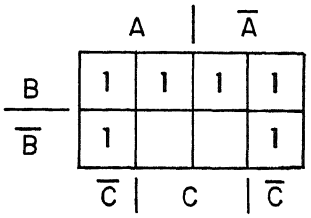
6. Half of the squares are assigned to each variable, and the other half to its complement. Each variable overlaps every other variable, and every complement but its own.

7. Plot one term at a time on the diagram. On an eight-square Veitch diagram a one-variable term (A) occupies four squares; a two-variable term ($A\overline{B}$) occupies two squares; and a three-variable term ($A\overline{B}C$) occupies one square.

Thus, $A\overline{B} + A\overline{B}\overline{C}$ would be plotted:



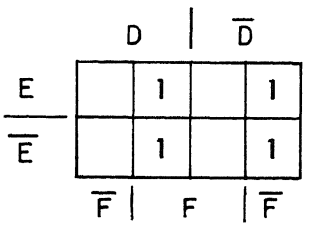
8. The plots for two or more terms may overlap. $B + \overline{C}$ would be plotted:



9. To simplify an expression, describe the plot with as few terms as possible. Look for patterns of plotted squares in this order:

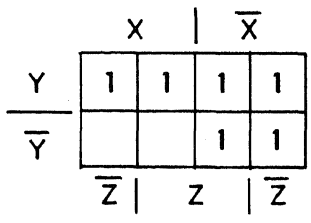
1. Four plotted squares described by a one-variable term
2. Two plotted squares described by a two-variable term
3. One plotted square described by a three-variable term

$DEF + \overline{D}\overline{F} + D\overline{E}\overline{F}$ is plotted:



The simplified expression is $DF + \overline{D}\overline{F}$.

10. Describe the plot with as few variables as possible. $\overline{X}$ and Y overlap in the following plot. By using two squares twice, the plot is described as $\overline{X} + Y$. This is preferable to $\overline{X} + XY$ or $Y + \overline{X}Y$.



A2-21. 1. $K + \overline{L}$

2. $\overline{X} + \overline{Z}$

3. $M\overline{P} + M\overline{N} + \overline{N}\overline{P}$

4. $B + C$

The simplified expression is extracted from the 16-square Veitch diagram in the same manner as it was in the 8-square Veitch diagram. Look for the following patterns:

1. Eight plotted squares describe a one-variable term
2. Four plotted squares describe a two-variable term
3. Two plotted squares describe a three-variable term
4. One plotted square describes a four-variable term

Patterns can be spotted quickly by recognizing that patterns are formed by adjacent squares and by squares at opposite ends of rows.

The Harvard Chart

Another technique of interest is the Harvard chart. Its use is demonstrated in figure 2-13. A properly constructed Harvard chart will list all variables under consideration plus the complements of these variables and all possible items which could be derived as a result of combining two or more of these variables and/or complements.

1	2	3	4	5	6	7	
A	B	C	AB	AC	BC	ABC	
~~$\overline{A}$~~	~~$\overline{B}$~~	~~$\overline{C}$~~	~~$\overline{A}\overline{B}$~~	~~$\overline{A}\overline{C}$~~	~~$\overline{B}\overline{C}$~~	~~$\overline{A}\overline{B}\overline{C}$~~	Row 1
~~$\overline{A}$~~	~~$\overline{B}$~~	~~C~~	~~$\overline{A}\overline{B}$~~	~~$\overline{A}$C~~	~~$\overline{B}$C~~	~~$\overline{A}\overline{B}$C~~	Row 2
~~$\overline{A}$~~	(B)	~~$\overline{C}$~~	~~$\overline{A}$B~~	~~$\overline{A}\overline{C}$~~	~~B$\overline{C}$~~	~~$\overline{A}$B$\overline{C}$~~	Row 3
~~$\overline{A}$~~	(B)	~~C~~	~~$\overline{A}$B~~	~~$\overline{A}$C~~	~~BC~~	~~$\overline{A}$BC~~	Row 4
~~A~~	~~$\overline{B}$~~	~~$\overline{C}$~~	~~A$\overline{B}$~~	~~A$\overline{C}$~~	~~$\overline{B}\overline{C}$~~	~~A$\overline{B}\overline{C}$~~	Row 5
~~A~~	~~$\overline{B}$~~	~~C~~	~~A$\overline{B}$~~	(AC)	~~$\overline{B}$C~~	~~A$\overline{B}$C~~	Row 6
~~A~~	(B)	~~$\overline{C}$~~	~~AB~~	~~A$\overline{C}$~~	~~B$\overline{C}$~~	~~AB$\overline{C}$~~	Row 7
~~A~~	(B)	~~C~~	~~AB~~	(AC)	~~BC~~	~~ABC~~	Row 8

Figure 2-13.—The Harvard chart.

Simplify

$$f = AB\overline{C} + ABC + \overline{A}BC + \overline{A}B\overline{C} + A\overline{B}C$$

1. Draw a line through all rows the terms of which DO NOT appear in the equation being simplified. (In this case none of the terms in rows 1, 2, and 5 appear in the equation and, hence, these rows will be lined out.)

2. Starting with the left column (column 1) and working to the right, cross out all terms which were lined out in step 1. In column 1, for example, $\overline{A}$ was lined out in rows 1 and 2 as a result of step 1, and A in row 5. Hence, all terms in column 1 will be lined (crossed) out.

3. In column 2 only the $\overline{B}$ terms are lined out as a result of steps 1 and 2. The terms not lined out (in this case the B terms) should be circled for easy identification as part of the final answer.

4. Working to the right of column 2, all terms containing B, in the rows where B was encircled as a result of step 3, should be lined out.

5. In column 5, the AC terms are not lined out as a result of previously executed steps and should be circled for easy identification as part of the final answer.

6. Working to the right of column 5, all terms containing AC, in the rows where AC was encircled as a result of step 5, should be crossed out.

7. As a result of previously executed steps, all terms in columns 6 and 7 are lined out and the process ends. Only B and AC are left. The final answer, therefore, is $f = B + AC$.

The Harvard chart may be used to simplify Boolean expressions containing any number of variables. It is more practical for use than the Veitch diagram when simplifying expressions containing more than four variables.

SUMMARY OF BOOLEAN ALGEBRA

The information in this chapter is based upon the assumption that most quantities have only two possible states, either true or false. Boolean is a description of the input conditions necessary to get a desired output from a logic circuit.

TRUTH TABLE–Used to summarize the logic circuit; can be used to describe the input conditions necessary to obtain a desired output from a logic gate.

BOOLEAN EXPRESSION–Written description of the logic in a circuit, e.g. $\overline{AB}$.

VINCULUM–The straight horizontal line or lines which are placed above a letter or letters in a Boolean expression to indicate negation and can serve as a sign of grouping.

$$\overline{\overline{A}} = A$$

$$\overline{AB} = \overline{A} + \overline{B}$$

$$\overline{(A + B)} = \overline{A}\,\overline{B}$$

LOGIC AND GATE–In chapter 1, it was pointed out that the AND gate is configured so that each input must be a true or binary 1 to get a true or binary 1 output.

The AND function is indicated by placing a dot (·) between letters or simply by grouping the terms together.

$$A \cdot B = AB$$

LOGIC OR GATE–The logic OR gate function is indicated by placing a plus sign (+) between terms. It was shown in Logic Circuits that only one input need be true (1) to get a true (1) output.

LOGIC NAND GATE–The logic NAND gate function can be expressed as an AND gate with an inverter on its output. The vinculum, or bar, is used as a grouping symbol to indicate this inversion.

LOGIC NOR GATE–The logic NOR gate is a basic OR gate with an inverter on its output.

LOGIC NOT FUNCTION–The NOT function is an inverter which is placed either on the input or the output side of a logic gate.

An example of this would be to place the Inverter on the output of the AND gate to form the NAND gate, or to place the Inverter on the output of an OR gate to form the NOR gate.

Another purpose of the Inverter is to invert the input signal to any logic gate.

OUTPUT EXPRESSION OF A LOGIC DIAGRAM–The following facts should be remembered when diagraming the output expression of a logic diagram.

- A logic diagram is composed of two or more logic symbols.
- If a logic symbol is at the extreme left of a diagram, its inputs are single letters.
- Parentheses are used to indicate grouping.
- The vinculum is used to group the output expressions that have been inverted.

BOOLEAN LAWS AND THEOREMS

The LAW OF IDENTITY states that any expression is equal to itself, that is, A = A.

The COMMUTATIVE LAW states that when logic symbols are ANDed or ORed together the order in which they are written does not affect their value, e.g. ABC = CAB.

The ASSOCIATIVE LAW states that expressions such as A(BC) or A + (B + C) can be simplified by rewriting and rediagraming.

$$A(BC) = ABC$$

$$A + (B + C) = A + B + C$$

The IDEMPOTENT LAW states that combining a quantity with itself either by logical addition or logical multiplication will result in a logical sum or product that is the equivalent of the quantity. This can be stated as $A + A + A = A$ or $AAA = A$.

The LAW OF DOUBLE NEGATION is used to aid simplification in logic expressions which have a number of vincula contained in them. Remember that if there is an even number of vincula, write that portion of the expression as a non-NOT function; and if there is an odd number of vincula, that portion of the expression will be written as a NOT function, e.g. $\overline{\overline{AB}} + \overline{\overline{C}} = \overline{AB} + C$.

The COMPLEMENTARY LAW may be restated as the logic AND and OR gate functions in an expressionary form, e.g. $A\overline{A}$ = false (0) or $A + \overline{A}$ = true (1). In other words an AND function requires all inputs be true to get a true output and an OR function requires that only one input be true to get a true output.

The LAW OF INTERSECTION states that if one input to a two-input AND gate is already true, then the output will depend upon the state of the other input only.

The LAW OF UNION is the same as the law of intersection except that it applies to the OR gate function. That is, if one input is already false, the only way to get a true output is if the other input is true.

DEMORGAN'S THEOREM is concerned with NAND and NOR functions. With this theorem it is possible to split or join vincula. The first part of the theorem deals with NAND functions and states that $\overline{AB} = \overline{A} + \overline{B}$. The second part of the theorem deals with NOR functions and states that $\overline{A + B} = \overline{A}\,\overline{B}$.

The DISTRIBUTIVE LAW is an application of normal algebra in that it states that $A(B + C) = AB + AC$. This law is used to manipulate a logic expression so that one of the other laws can be used to simplify it.

The LAW OF ABSORPTION is another of the laws which is used to manipulate a logic expression in order to simplify it. The law of absorption states that $A(A + B) = A$ or $A + AB = A$. This, in effect, says that anytime that you have an A you will get an A output.

The LAW OF COMMON IDENTITIES is a law which governs the most frequently occurring Boolean expressions that would normally be simplified by applying a combination of other Boolean laws. Once these identities are learned, they increase the speed of simplification. The law of identities states that anytime the expression $A(\overline{A} + B) = AB$ or $A + \overline{A}B = A + B$ appears it can immediately be simplified to AB or A + B respectively without going through the process of using the distributive or complementary laws, or the law of union to simplify.

The VEITCH DIAGRAM is used in finding the simplest logic equation needed to express a given function. This simplification method is based on the fact that a Boolean expression has two states that are complementary to each other. Any number of variables may be plotted on a Veitch diagram, though the diagrams become difficult to use when more than four variables are involved.

The HARVARD CHART is used to simplify equations with more than four variables.

Table 2-2.—Boolean laws and theorems

1.	Law of Identity	$A = A$ $\overline{A} = \overline{A}$
2.	Commutative Law	$AB = BA$ $A + B = B + A$
3.	Associative Law	$A(BC) = ABC$ $A + (B + C) = A + B + C$
4.	Idempotent Law	$AA = A$ $A + A = A$
5.	Double Negative Law	$\overline{\overline{A}} = A$
6.	Complementary Law	$A\overline{A} = 0$ $A + \overline{A} = 1$
7.	Law of Intersection	$A \cdot 1 = A$ $A \cdot 0 = 0$
8.	Law of Union	$A + 1 = 1$ $A + 0 = A$
9.	DeMorgan's Theorem	$\overline{AB} = \overline{A} + \overline{B}$ $\overline{A + B} = \overline{A}\,\overline{B}$
10.	Distributive Law	$A(B + C) = AB + AC$ $A + BC = (A + B)(A + C)$
11.	Law of Absorption	$A(A + B) = A$ $A + AB = A$
12.	Law of Common Identities	$A(\overline{A} + B) = AB$ $A + AB = A + B$

CHAPTER 3

NUMBER SYSTEMS

People and computers, since they do not speak the same language, need methods of interpretation or conversion to understand each other. Humans generally speak in words or decimal numbers. A computer, on the other hand, understands only the coded electronic pulses that represent digital or binary information. This chapter will describe the basic arithmetic operations (addition, subtraction, multiplication, and division) as they are performed in binary, octal, and hexadecimal number systems. Techniques for converting between these systems and the decimal number system will also be discussed.

A number system is any set of symbols or characters used for the purposes of counting and performing arithmetic operations. These symbols or characters are commonly referred to as digits. Each numbering system will have its own set of digits. A study of the table in figure 3-1 shows that certain symbols are common to several different numbering systems. For example, the digits of the binary system are common to all numbering systems currently being used in digital data processing. Because they have common digits and because basic operations are the same for all systems, certain relationships may be established among the various number systems and conversions from one system to another may be easily accomplished.

The binary number system is more readily adopted to electronic digital data equipment because it is the most basic number system. It contains only two symbols, 1 and 0. These can easily be mechanized by switches, relays, or transistor logic. As shown in earlier chapters, the logic gate has a high, logic 1, or true state and a low, logic 0, or false state. Since these states can represent the binary number system, it is a very simple matter to use these logic devices to adopt the binary number system to an electronic data processing machine.

Q3-1. What symbols are used to represent numbers in the binary number system?

Early number systems were based on the tally system of counting, in which a tick mark or separate symbol was used to indicate each object counted. Because such systems made computing extremely difficult, they were limited in use to taking measurements and keeping records. In early civilizations, the lack of an adequate number system was probably a major factor limiting scientific development.

The discovery and acceptance of two basic concepts greatly simplified mathematical computations and led to the development of modern number systems. These concepts are (1) the principle of positional value, and (2) the use of zero.

In the principle of positional value, a digit is assigned a value based on two factors. These are (1) the digit's basic value, that is, the number of units it represents by itself, and (2) a weighting value, which is determined by the digit's position within a given number. For example, in each of the decimal numbers 123, 132, and 312 the digit 3 has a different value. In the first number the 3 has its basic value (3); in the second number the 3 has a value of 30 (3 x 10); and in the third number the 3 has a value of 300 (3 x 10 x 10).

A3-1. 1 and 0

BINARY	OCTAL	DECIMAL	HEXADECIMAL
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
10000	20	16	10
10001	21	17	11
10010	22	18	12
10011	23	19	13
10100	24	20	14
10101	25	21	15
10110	26	22	16
10111	27	23	17
11000	30	24	18
11001	31	25	19
11010	32	26	1A
11011	33	27	1B
11100	34	28	1C
11101	35	29	1D
11110	36	30	1E
11111	37	31	1F
100000	40	32	20
100001	41	33	21
100010	42	34	22
↓	↓	↓	↓

Figure 3-1.—A comparison of number system symbols.

Sometimes a position within a given number will not have a value assigned. However, should this position merely be omitted, then there would be no way to distinguish between numbers such as 505 and 55. Thus, the zero is used to signify that a particular digit position within a given number has no value assigned, or in the case of a single digit number that no units have been counted.

Now, consider the following definitions which pertain to all number systems.

Unit—a single object or thing.

Number—an arbitrary symbol (called a numeral) or group of symbols representing a sum of units.

Number system—a method of indicating the sum of units counted.

Radix or base—the number of different symbols a number system uses including the zero.

Quantity—a number of units (implies both a number and a unit).

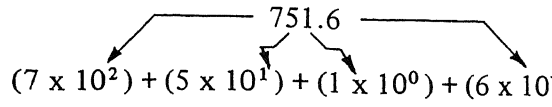
Modulus—the total number of different numbers or stable conditions that a counting device can indicate. For example, the odometer on most automobiles has a modulus of 100,000 since it indicates all numbers from 00,000 to 99,999. The modulus of the hour hand on most watches is 12 and that of the minute hand is 60.

Q3-2. The digit's basic value and a weighting value are the two factors that determine the _____.

POSITIONAL NOTATION

Positional notation is the standard shorthand form of writing numbers. The value of a particular digit depends not only on the digit value, but also on the position of the digit within the number. Consequently, the decimal number 751.6 is the shorthand form of the quantity seven hundred fifty-one and six-tenths.

What is actually stated by the shorthand form is best illustrated by the following example:



A quantity can be expressed in the positional notation (standard shorthand) form in any numbering system. The general form is:

$$Q = (d_n \times r^n) + \ldots\ldots + (d_2 \times r^2) + (d_1 \times r^1) + (d_0 \times r^0) + (d_{-1} \times r^{-1}) + (d_{-2} \times r^{-2}) + \ldots\ldots + (d_{-n} \times r^{-n})$$

Where:

Q is the quantity expressed in positional notation form;

r is the radix or base of the number system in use;

d is any one of the various symbols or characters used by the system; and

n (both exponential and subscript) is a decimal number designating the symbol's or character's position within the number.

This is an impressive looking equation; however, using an example in the decimal number system will clarify it greatly. The number 751.6 will be used for demonstration purposes.

$$Q = (d_2 \times r^2) + (d_1 \times r^1) + (d_0 \times r^0) + (d_{-1} \times r^{-1})$$

$$Q = (7 \times 10^2) + (5 \times 10^1) + (1 \times 10^0) + (6 \times 10^{-1})$$

$$Q = 700 + 50 + 1 + .6$$

$$Q = 751.6$$

Note that the use of the radix point (known as the decimal point in the decimal system) is not required in the general expression. However, when the shorthand form is used the radix point must be placed between the $d_0 \times r^0$ and $d_{-1} \times r^{-1}$ values in order to separate the whole and fractional portions of the number. Further, when the shorthand form is used to express a whole number, the use of the radix point is understood. That is, when a number is written in shorthand form and the radix point is omitted, the number is understood to be a whole number or integer and the radix point should be placed immediately to the right.

Q3-3. What is the positional notation for the decimal number 67.42?

THE RADIX

Every number system has a radix, or base. When the radix (r) is 10, the decimal system is indicated; when the r is 8, the octal system is indicated; and when the r is 2, the binary system is indicated. The division between integers (whole numbers) and fractions is recognized by the position of the radix point. The following are additional characteristics of the radix.

1. The radix of a numbering system is equal to the number of the different characters which are necessary to indicate all the various values a digit may represent. For example, the decimal system, with a radix of 10, has 10 digits representing the values 0 through 9.

2. The value of the radix is always one unit greater than the largest basic character being used. This is because the radix is equal to the number of characters, whereas the characters themselves start from zero. Thus, the octal system (discussed later) has a radix of 8 and uses digits 0 through 7.

3. The positional notation does not, in itself, indicate the radix. The symbol "312" could represent a number written in the quartic (base 4), octal, or decimal system, or in any system having a radix of 4 or more. To avoid confusion, numbers written in systems other

A3-2. Principle of positional value

A3-3. $(6 \times 10^1) + (7 \times 10^0) + (4 \times 10^{-1}) + (2 \times 10^{-2})$

---

than the decimal system should have the radix noted as a subscript.

$$315.72_8$$

The radix subscript is always written as a decimal (base 10) number.

4. Any number can easily be multiplied or divided by the radix of its number system. In decimal notation, to multiply a number by 10, move the decimal (radix) point one position to the right of its former position.

$$\begin{array}{r} 34.564 \\ \times \quad 10 \\ \hline 345.64 \end{array}$$

Often overlooked is the fact that the radix point could remain stationary and the digits be moved.

$$\begin{array}{r} 034.564 \\ \times \quad 10 \\ \hline 345.640 \end{array}$$

In the same fashion, a binary number may be multiplied by 2 (10_2) merely by moving the binary (radix) point one position to the right, or by shifting the number one position to the left while the binary point remains fixed.

$$10101.01_2 \times 10_2 = 101010.1_2$$

To divide a number by 10, move the decimal (radix) point one digit to the left of its former position, or move the digits one digit space to the right relative to the radix point.

$$\frac{34.564}{10} = 3.4564 \text{ or } 34.564 \div 10 = 3.4564$$

Q3-4. The radix of a number system is 6. What digits are used in the number system?

Q3-5. Multiplication of a number by its radix will move the radix point one position to the _____.

WEIGHTING VALUES

By convention, weighting values will always be arranged in the same manner. That is, regardless of the numbering system in use, the highest weighting value will be on the extreme left, and the lowest on the extreme right. Thus, positional coefficients (or weighting values) increase in value from right to left.

Q3-6. Which digit in the number 123.4 has the largest weighting value? The lowest weighting value?

The radix point is always the starting point when determining weighting values or positional coefficients. The position immediately to the left of the radix point is the units position and, regardless of the numbering system in use, will have a weighting value or positional coefficient of one. That is, any digit (including the zero) appearing in this position will be multiplied by 1. The weighting value of each subsequent position to the left of the units position will be increased by a power of the radix of the numbering system in use. Conversely, the weighting value of the first and each subsequent position to the right of the radix point will be decreased by a negative coefficient (or power) of the radix of the numbering system in use.

As illustrated in figure 3-2, each number will have a "most significant digit" (MSD) and a "least significant digit" (LSD). In a whole number the leftmost digit other than zero is termed the MSD. This is because it will be multiplied by the highest value positional coefficient and, therefore, will have the most effect on the results of any computations (addition, subtraction, multiplication, and division) involving the number. Conversely, the rightmost digit, whether it is a 1 or a 0, is

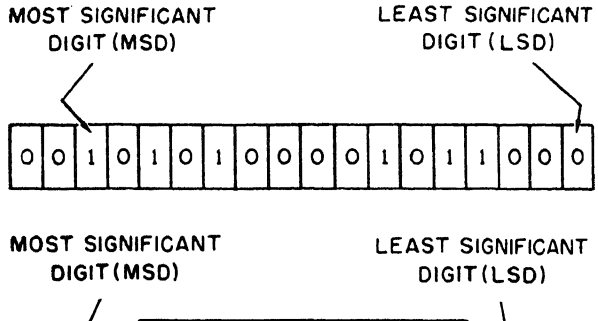


Figure 3-2.—MSD and LSD.

termed the LSD because it will be multiplied by the lowest value positional coefficient and will have the least effect on the results of any computations involving the number. In a whole number the LSD always appears immediately to the left of the radix point (in the units position) and, hence, will have a positional coefficient of one.

When dealing with fractional numbers the situation will be a bit different. The MSD will appear immediately to the right of the radix point and the LSD will be the rightmost digit other than zero.

For example: .0110

MSD LSD

In this case any digit, including zero, appearing in the MSD position will be multiplied by the first negative coefficient or power of the radix of the numbering system in use.

If mixed numbers are used, the situation is again altered. The digit (other than zero) farthest to the left of the radix point will be the MSD and the digit (other than zero) farthest to the right of the radix point will be the LSD.

For example: 0110.0110

MSD LSD

Hence, the MSD will be multiplied by a positive coefficient and the LSD by a negative coefficient.

Referring again to figure 3-2, both of th numbers are considered to be whole numbers This is true because, without any indication t the contrary, it must be assumed that the radi point will, in both cases, appear immediately t the right of the rightmost digit.

Consider now the number 43,096. In thi number the MSD is 4. When multiplied by it positional coefficient of 10^4 it will have a valu of 40,000 and, hence, will have a considerabl effect on any computations involving th number. The LSD is 6 and, since it appears i the units position, will have a positiona coefficient of 10^0, or 1. Since 6 is such a smal value when compared to 40,000, its effect o any computations involving the number will b relatively minor.

Q3-7. What are the MSD and LSD o 2563.987?

BASE 2, 8, 10, AND 16 RELATIONSHIPS

The rules for counting numbers written in system of positional notation are the same fo every radix. The octal system is used in th following example to illustrate these rules.

1. Starting from zero, add 1 to the leas significant digit until all basic characters hav been used:

0, 1, 2, 3, 4, 5, 6, 7, ...

2. Since 7 is the largest character in th system, a larger number requires two digit Start the series of two-digit numbers with zer as the least significant digit and a 1 to the left o that zero:

. . . 6, 7, 10, 11, 12, 13, 14, 15, 16, 17, . . .

3. Whenever any digit reaches its maximu value (7, in this case), replace it with zero an add 1 to its next more significant digit:

. . . 16, 17, 20, 21, . . 26, 27, 30, 31, . . . 66, 6 70, 71, . . .

A3-4. 0 through 5

A3-5. right

A3-6. 1 and 4, respectively

A3-7. 2 and 7, respectively

---

4. When two or more consecutive digits reach the maximum value, replace them with zeros and add 1 to the next more significant digit:

. . . 76, 77, 100, 101, . . . 176, 177, 200, . . . 776, 777, 1000, . . .

NOTE: The symbol 10 always represents the radix in its own system. This is true because the radix is one unit larger than the largest character, and by the rules of counting, this value is written as 10.

For example:

Binary 10 = 2 (the radix of the binary system)
Octal 10 = 8 (the radix of the octal system)
Decimal 10 = 10 (the radix of the decimal system)
Hexadecimal 10 = 16 (the radix of the hexadecimal system)

Decimal System

Since the decimal system (also known as the Hindu-Arabic system) uses 10 symbols or digits (fig. 3-1), it has a radix or base of 10. This system is thought to have evolved and become commonly used as a result of our having 10 fingers (digits).

Since this system is used almost the world over, basic mathematical computations performed by a person in one country are easily understood by a person in another country. Because of its common usage and because of its relationship with other number systems, this system also serves as a basis for discussion of other number systems.

Binary System

The simplest possible number system is based on powers of 2 and is known as the binary system. The table in figure 3-1 illustrates the relation between this system, the decimal system, and other systems commonly used in present day computer systems.

By a convenient coincidence, the two binary conditions (1 and 0) can easily be represented by many electrical/electronic components. For example, the binary 1 state may be indicated when the component is conductive and the 0 state may be indicated when the component is nonconductive. The reverse will work equally well; that is, the nonconducting state of a component can be represented by the binary 1 condition and a conducting state by the binary 0 condition. Both procedures are used in digital computer applications and frequently within the same computer system. Numerous devices may be used to provide a representation of binary conditions. These include switches, transistors, relays, and diodes.

The quantity represented using binary characters (or the characters in any numbering system) cannot be determined without knowing the positional weighting value of each character (digit). The positional values of binary characters from 2^0 to 2^9 (1_{10} to 512_{10}) are illustrated in figure 3-3.

Consider the following hypothetical example.

A number of flip-flops are arranged in a chain to form a "register." A flip-flop is an electronic device which has two stable states, set (logic 1) and clear (logic 0). (Flip-flops are covered in detail in NAVEDTRA 10088-B, *Digital Computer Basics*.) These devices are used in digital equipment to form registers, which are simply groups of flip-flops arranged to hold binary numbers. By gating pulses into the registers, certain of these flip-flops may be forced to assume a set (logic 1) state and others a clear (logic 0) state. In figure 3-3 a register with 10 flip-flops is shown, and the first binary number shown is 0000010101. Flip-flops are numbered by the positional value of the exponent; that is, flip-flop 4 represents 2^4. Flip-flops 4, 2, and 0 are set, or in the 1 state. The other flip-flops are clear, or in the 0 state.

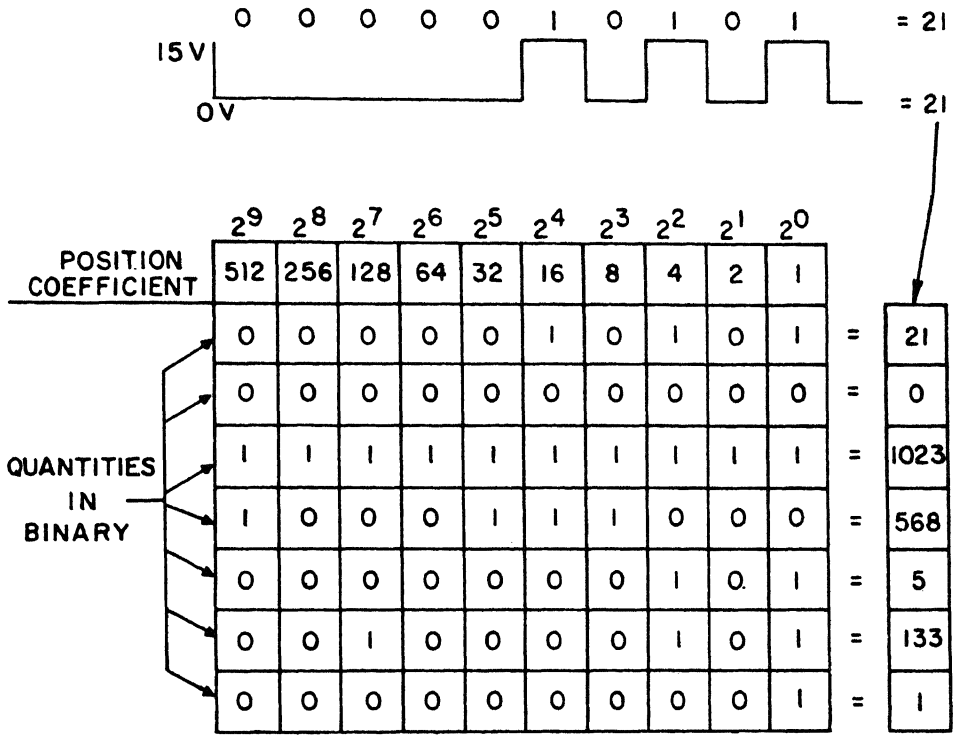


Figure 3-3.—Positional weighting values.

The value of this number can be determined by simply summing the positional values as indicated by the 1s in figure 3-3. By this procedure, the top number yields:

$$(1 \times 2^4) + (1 \times 2^2) + (1 \times 2^0) = 21$$

$$(1 \times 16) + (1 \times 4) + (1 \times 1) = 21$$

Thus, $0000010101_2 = 21_{10}$, and it is now known that the flip-flop chain or register is storing the binary equivalent of decimal 21.

The value of each of the other binary numbers in figure 3-3 is determined in the same manner.

Octal System

The octal system has eight distinct characters (fig. 3-1), thus its radix is 8. The octal system is quite useful as an accessory to the binary system, because eight is an integral power of two ($8 = 2^3$). One octal digit has a value equivalent to that of three binary digits and vice versa.

Octal to Binary			Binary to Octal		
2	2	5_8	010	010	101_2
010	010	101_2	2	2	5_8

NOTE: The binary equivalent of the MSD (2) uses the symbols 010 even though the leftmost zero does not change the value of the number converted.

This relationship expedites the programming of digital machines. The octal system may be used in place of the more cumbersome binary system, which is the actual language of digital machines. The conversion from octal to binary, and vice versa, is, then, a simple process which

may be accomplished at any point in the system as desired.

Q3-8. How many binary digits are needed to express two octal digits?

Hexadecimal System

The hexadecimal system has a radix of 16. The 10 digits of the decimal system and the first six letters of the alphabet are the symbols most commonly used to represent the 16 digits of the hexadecimal system. (See fig. 3-1.) Sixteen, like eight, is an integral power of two ($16 = 2^4$). Thus, one hexadecimal digit has a value equivalent to that of four binary digits and vice versa.

Hexadecimal to Binary			Binary to Hexadecimal		
E	2	5_{16}	1110	0010	0101_2
1110	0010	0101_2	E	2	5_{16}

Despite the relationship between the binary and the hexadecimal systems, the hexadecimal system finds only limited usage in military digital data processing systems. One such usage is in the IBM 360 system. Here the organization of the central processor memory is such that the hexadecimal system is used to allow easier memory access. The hexadecimal system is being used more and more in modern microcomputers in civilian environments.

Q3-9. How many binary digits are needed to express three hexadecimal numbers?

The information contained in the rest of this chapter describes arithmetic techniques. These techniques are somewhat difficult to teach in a formal classroom atmosphere. They become even more difficult when described in a text and a correspondence course. It is strongly recommended that the reader proceed slowly through the rest of this material. Ensure that each of the processes is fully understood and then work carefully through each of the examples before attempting the correspondence course.

ARITHMETIC OPERATIONS

There are four basic arithmetic operations: addition, subtraction, multiplication, and division. In general, the rules for performing such operations will be the same regardless of which number system is being used. Developing some skill in performing the basic operations will make the methods used to convert from one number system to another easier to understand. Conversion methods are described in the final portion of this chapter.

Addition

Addition is a form of counting where one quantity is added to another. The following definitions identify the basic terms of addition.

Addend—a number to be added to a preceding number.

Augend—the quantity to which an addend is added.

Sum—the result of an addition (the sum of 5 and 7 is 12).

Carry—a carry is produced when the sum of two or more digits in a column equals or exceeds the base of the number system in use. To handle the carry, that is, to handle the two digit number generated when a carry is produced, the lower order digit becomes the sum of the column being added and the higher order digit (the carry) is added to the next higher order column. For example, in the decimal system:

```
  1  Carry
 15  Augend
+ 6  Addend
 21  Sum
```

The sum of 5 and 6 is 11. The lower order 1 becomes the sum of the lower order column, and the upper order 1 (the carry) is added to the

+	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10	11
3	3	4	5	6	7	8	9	10	11	12
4	4	5	6	7	8	9	10	11	12	13
5	5	6	7	8	9	10	11	12	13	14
6	6	7	8	9	10	11	12	13	14	15
7	7	8	9	10	11	12	13	14	15	16
8	8	9	10	11	12	13	14	15	16	17
9	9	10	11	12	13	14	15	16	17	18

A (row) C (table) B (column)

Figure 3-4.—Decimal addition table.

upper order column producing a sum of 2 for that column. The sum of 15 and 6 is, therefore, 21.

Although the rules for addition are basically the same regardless of the number system being used, each number system, because it has a different number of digits, will have a unique digit addition table. These addition tables will be described as the adding process for each number system discussed.

A decimal addition table is shown in figure 3-4. The numbers in row A and column B may represent either the addend or the augend. If the numbers in A represent the augend then the numbers in B must represent the addend and vice versa. The sum of A + B is located at the point in array C where the selected A row and B column intersect. This chapter is written under the assumption that the reader is competent in decimal arithmetic operations; therefore, no examples will be shown of decimal arithmetic.

A binary addition table is illustrated in figure 3-5. Notice that the binary system has only two digits, 0 and 1. The binary carry, when generated, will be raised in value by the first power of 2. For example, when we add 1 + 1, the sum is 2 (2_{10}). However, the symbol 2 does not exist in the binary system, so we write 10_2 which is the binary equivalent of decimal 2. When written in positional notation form 10_2 is $(1_2 \times 2^1) + (0_2 \times 2^0)$. The 1 which represents the carry is, therefore, raised in value by the first power of 2.

Example: Add 1011_2 and 1101_2.

Solution: Write

$$\begin{array}{r} 1011_2 \text{ Augend} \\ +\,1101_2 \text{ Addend} \\ \hline \end{array}$$

As previously noted, the sum of 1 and 1 is 2, which cannot be expressed as a single digit in binary. Therefore,

$$\begin{array}{r} 1 \quad \text{Carry} \\ 1011_2 \text{ Augend} \\ +\,1101_2 \text{ Addend} \\ \hline 0_2 \end{array}$$

+	0	1
0	0	1
1	1	10

A (row) C (table) B (column)

Figure 3-5.—Binary addition table.

A3-8. 6

A3-9. 12

The following steps, with the carry indicated, show the completion of the addition.

Previous Carry Used

$1\not{1}$ Carry
1011_2 Augend
$+\,1101_2$ Addend
00_2

Previous Carries Used

$1\not{1}\not{1}$ Carry
1011_2 Augend
$+\,1101_2$ Addend
000_2

1011_2 Augend
$+\,1101_2$ Addend
11000_2 Sum

Notice in the last step that three 1s were added. However "3" cannot be expressed as a single digit in binary, so we write

$$11_2$$

Example: Add 100_2, 101_2, and 111_2.

Solution: Write

100_2 Augend
101_2 Addend
$+\,111_2$ Addend

1 Carry
100_2 Augend
101_2 Addend
$+\,111_2$ Addend
0_2

Previous Carry Used

$1\not{1}$ Carry
100_2 Augend
101_2 Addend
$+\,111_2$ Addend
00_2

100_2 Augend
101_2 Addend
$+\,111_2$ Addend
10000_2 Sum

Notice that in the last step four 1s were added to produce a total of "4." There is no single digit in the binary system to represent "4." We have to write the binary equivalent which is 100_2.

The octal system has the digits 0, 1, 2, 3, 4, 5, 6, and 7. When an addition carry is made, the carry is raised in value by the first power of eight. The octal addition table is shown in figure 3-6. When adding 7 and 6, the sum is expressed in base 8 as one group of 8 and five groups of 1.

Write

7_8 Augend
$+\,6_8$ Addend
15_8 Sum

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

A, B, C

Figure 3-6.—Octal addition table.

Example: Add 765_8 and 675_8.

Solution: Write

$$\begin{array}{r} 765_8 \text{ Augend} \\ +\,675_8 \text{ Addend} \\ \hline \end{array}$$

$$\begin{array}{r} 1 \quad \text{Carry} \\ 765_8 \text{ Augend} \\ +\,675_8 \text{ Addend} \\ \hline 2_8 \end{array}$$

Previous Carry Used

$$\begin{array}{r} 1\gamma \quad \text{Carry} \\ 765_8 \text{ Augend} \\ +\,675_8 \text{ Addend} \\ \hline 62_8 \end{array}$$

$$\begin{array}{r} 765_8 \text{ Augend} \\ +\,675_8 \text{ Addend} \\ \hline 1662_8 \text{ Sum} \end{array}$$

The hexadecimal or base 16 addition table i
shown in figure 3-7. In this table digits havin

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	α
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	γ
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	

β

Figure 3-7.–Hexadecimal addition table.

values greater than 9 are represented by the letters A, B, C, D, E, and F. Because the letters A, B, and C are being used to represent digits, Greek letters are now used to designate the various parts of the table. Alpha (α) is used to designate the rows, beta (β) to designate the columns, and gamma (λ) to designate the array. In this system, when an addition carry is made, the carry is raised in value by the first power of 16.

Example: Add $3A9_{16}$ and $E86_{16}$.

Solution: Write

$$\begin{array}{rl} 3A9_{16} & \text{Augend} \\ +\ E86_{16} & \text{Addend} \\ \hline F_{16} & \end{array}$$

$$\begin{array}{rl} 1 & \text{Carry} \\ 3A9_{16} & \text{Augend} \\ +\ E86_{16} & \text{Addend} \\ \hline 2F_{16} & \end{array}$$

$$\begin{array}{rl} 3A9_{16} & \text{Augend} \\ +\ E86_{16} & \text{Addend} \\ \hline 122F_{16} & \text{Sum} \end{array}$$

Example: Add $BC2_{16}$ and EFA_{16}.

Solution: Write

$$\begin{array}{rl} BC2_{16} & \text{Augend} \\ +\ EFA_{16} & \text{Addend} \\ \hline C_{16} & \end{array}$$

$$\begin{array}{rl} 1 & \text{Carry} \\ BC2_{16} & \text{Augend} \\ +\ EFA_{16} & \text{Addend} \\ \hline BC_{16} & \end{array}$$

$$\begin{array}{rl} BC2_{16} & \text{Augend} \\ +\ EFA_{16} & \text{Addend} \\ \hline 1ABC_{16} & \text{Sum} \end{array}$$

Subtraction

The following definitions identify the basic terms needed to understand subtraction operations.

Subtract—to take away, as a part from the whole or one number from another.

Minuend—the number from which another number is to be subtracted.

Subtrahend—the quantity to be subtracted.

Remainder or Difference—that which is left after subtraction.

Borrow—to take one from the minuend digit in the next higher order column and add the base of the number system in use to the minuend digit in the lower order column; the one from the higher order column, of course, being the equivalent of the base in the next lower order column.

Since the process of subtraction is the opposite of addition, the addition tables may be used to illustrate subtaction facts for the number system under discussion.

In addition

$$A + B = C$$

In subtraction the reverse is true, that is,

$$C - B = A$$

or

$$C - A = B$$

Thus, in subtraction the minuend is always found in array C and the subtrahend in either row A or column B. If the subtrahend is in row A, then the remainder will be in column B. Conversely, if the subtrahend is in column B, then the remainder will be in row A.

The decimal subtraction table is found in figure 3-4. To subtract 8 from 15, find 8 in either the A row or B column. Find where this row or column intersects with a value of 15 for C, then move to the remaining row or column to find the remainder.

This problem, when written in the familiar form of

15 Minuend
- 8 Subtrahend
7 Remainder

requires the use of the "borrow;" that is, 8 cannot be subtracted from 5 and leave a positive remainder. It is necessary to borrow the 1, which is really one group of 10. Then, one group of 10 plus five groups of 1 equals 15, and 15 minus 8 leaves a remainder of 7.

When subtracting in base 2, use the addition table in figure 3-5. To subtract 1_2 from 10_2, use the borrow.

10_2 Minuend
- 1_2 Subtrahend
1_2 Remainder

In this problem 1 cannot be subtracted from 0 and leave a positive remainder. Therefore, borrow the 1, which is really one group of 2. One group of 2 plus one group of 0 minus one group of 1 equals one group of 1. This is shown below.

$$(1_1 \times 2^1) + (0_0 \times 2^0) - (1_0 \times 2^0) = (1_0 \times 2^0)$$

Figure 3-5 provides the information necessary to verify the results of this subtraction.

Example: Subtract 11_2 from 101_2.

Solution: Write

101_2 Minuend
- 11_2 Subtrahend

Then, 1 from 1 is 0

101_2 Minuend
- 11_2 Subtrahend
0_2

Since 1 cannot be subtracted from 0 and leave a positive remainder, borrow the 1 from the left-hand column and add its value to the 0 in the second column before proceeding with the subtraction. Thus, 10_2 minus 1_2 equals 1_2 and may be written as

10 Borrow
$\not{1}01_2$ Minuend
- 11_2 Subtrahend
10_2 Remainder

Figure 3-6 contains the octal subtraction table. In this table

$$C - B = A$$

or

$$C - A = B$$

Example: Find the remainder when 6_8 is subtracted from 13_8.

Solution: In accordance with the table, if

$$C = 13_8$$

and

$$B = 6_8$$

then

$$C - B = A$$

$$13_8 - 6_8 = 5_8$$

Example: Subtract 326_8 from 432_8.

Solution: Write

432_8 Minuend
- 326_8 Subtrahend

Since 6 cannot be subtracted from 2 and leave a positive number, borrow 1 from the 3 in column two and add its value to the 2 in column one before proceeding with the subtraction. That is

$10_8 + 2_8 = 12_8$ and $12_8 - 6_8 = 4_8$.

Then, write

2 12 Borrow Result
$4\ \not{3}\ \not{2}_8$ Minuend
- $3\ 2\ 6_8$ Subtrahend
$1\ 0\ 4_8$ Remainder

The hexadecimal subtraction table is shown in figure 3-7. Subtraction in this system is the same as in the other systems previously discussed.

Example: Find the remainder when $39E_{16}$ is subtracted from $9C6_{16}$.

Solution: Write

$9C6_{16}$ Minuend
$-39E_{16}$ Subtrahend

Since E cannot be subtracted from 6 and leave a positive remainder, borrow 1 from the C in column two and add its value to the 6 in column one before proceeding with the subtraction.

B 16 Borrow Result
9 ~~C~~ ~~6~~ Minuend
-3 9 E Subtrahend
6 2 8 Remainder

Multiplication

Multiplication in any number system is performed in the same manner as in the decimal system. Each system has a unique digit multiplication table. These tables will be discussed with each system. The rows, columns, and arrays of these tables are labeled in the same fashion as the addition tables. Only the sign of operation and array values are different.

The following terms are commonly used in multiplication.

Multiplicand—the number that is to be multiplied by another number (called the multiplier).

Multiplier—the number by which another number is multiplied.

Product—the number resulting from the multiplication together of two or more numbers.

Partial product—the product obtained by performing one of the steps in a series of multiplications.

Factor—any of the elements, quantities, or symbols which, when multiplied together, form a product.

Row A (fig. 3-8) now contains either the multiplier or the multiplicand and the same applies for column B. However, should the multiplier be in row A, then the multiplicand must be in column B and vice versa. Array C contains the products.

$$A \times B = C$$

$$B \times A = C$$

NOTE: If one of the factors in a multiplication is 0, the product will be 0. For example: $5 \times 0 = 0$.

x	0	1	2	3	4	5	6	7	8	9	
											A
0	0	0	0	0	0	0	0	0	0	0	
1	0	1	2	3	4	5	6	7	8	9	
2	0	2	4	6	8	10	12	14	16	18	
3	0	3	6	9	12	15	18	21	24	27	
4	0	4	8	12	16	20	24	28	32	36	C
5	0	5	10	15	20	25	30	35	40	45	
6	0	6	12	18	24	30	36	42	48	54	
7	0	7	14	21	28	35	42	49	56	63	
8	0	8	16	24	32	40	48	56	64	72	
9	0	9	18	27	36	45	54	63	72	81	

B

Figure 3-8.—Decimal multiplication table.

Multiplication in the decimal system requires that the decimal digit multiplication and decimal digit addition tables be used in accordance with certain procedures or rules. These procedures or rules are well known and apply to direct multiplication in any number system. Figure 3-8 shows the decimal multiplication table.

The direct method of multiplication of decimal numbers is shown in the following example.

Example: Multiply 32 by 25.

Solution: Write

$$25 = 20 + 5$$

then

$$32(25)$$

$$= 32(20 + 5)$$

$$= 32(20) + 32(5)$$

$$= 640 + 160$$

$$= 800$$

The same problem, written as

32 Factor (multiplicand)

x 25 Factor (multiplier)

gives

$$32(5)$$

$$= 160 \text{ Partial Product}$$

then,

$$32(20)$$

$$= 640 \text{ Partial Product}$$

then,

$$160 + 640$$

$$= 800 \text{ Product}$$

The technique generally used is

```
  32 Factor
x 25 Factor
 160 Partial Product
 64  Partial Product
 800 Product
```

Notice that the 64 really represent 640 but the zero is omitted.

Example: Multiply 306 by 762.

Solution: Write

```
   306 Factor
x  762 Factor
   612 Partial Product
  1836 Partial Product
 2142  Partial Product
233,172 Product
```

NOTE: The order in which the terms are arranged has no effect on the product of a multiplication. Therefore, the term factor may, for the purpose of this text, be substituted for the terms multiplier and multiplicand.

Figure 3-9 shows the multiplication table for the binary system. This is the simplest set of facts of any of the number systems. As will be seen, the only difficulty in binary multiplication may be in the addition of the partial products.

Example: Multiply 101_2 by 1101_2.

Solution: Write

$$1101_2 \text{ Factor}$$
$$\times\ 101_2 \text{ Factor}$$

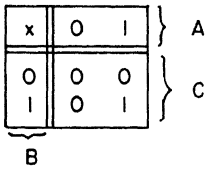


Figure 3-9.—Binary multiplication table.

The partial products and the product are as follows:

1101_2 Factor
x 101_2 Factor
1101 Partial Product
0000 Partial Product
1101 Partial Product
1000001_2 Product

As in the addition section, the problem that may be encountered in the addition of the partial products is what to carry. The following example will illustrate this problem.

Example: Multiply 1111_2 by 111_2.

Solution: Write

1111_2 Factor
111_2 Factor
1111 Partial Product
1111 Partial Product
1111 Partial Product

Add the partial products. The carry produced from the addition of column two is placed over column three. Column three is then added producing 100_2. A 0 is placed in column three of the product. The carry 10_2, generated from column three, is placed over columns four and five. When column four is added, the sum is 11_2. A 1 is placed in column four of the product and a carry of 1_2 is added to column five. Column five's partial products and carries are now added to produce 100_2. A 0 is placed in column five of the product and a carry of 10_2 is placed over columns six and seven. The addition of the partial product can now be completed without generating anymore carries and the product is now complete, 1101001_2.

Addition of the partial product

10	Carry generated by the sum of column five
1	Carry generated by sum of column four
10	Carry generated by sum of column three
1	Carry generated by sum of column two
1111	Partial Product
1111	Partial Product
1111	Partial Product
1101001_2	Product

x	0	1	2	3	4	5	6	7	
0	0	0	0	0	0	0	0	0	A
1	0	1	2	3	4	5	6	7	
2	0	2	4	6	10	12	14	16	
3	0	3	6	11	14	17	22	25	C
4	0	4	10	14	20	24	30	34	
5	0	5	12	17	24	31	36	43	
6	0	6	14	22	30	36	44	52	
7	0	7	16	25	34	43	52	61	

B

Figure 3-10.—Octal multiplication table.

The base 8 multiplication table for the octal system is shown in figure 3-10. Perhaps the most obvious way to find the product of two octal numbers is to convert the numbers to their decimal equivalents, find the product, and then convert the product to its octal equivalent. This process becomes time consuming and may even lead to confusion and error when performing a long series of octal multiplications. The best method, therefore, is to use the octal multiplication and addition tables when performing octal multiplication. In fact, if you are to perform octal multiplication on a regular basis, it would be advantageous to memorize these tables, thus eliminating the necessity of referring to them every time you perform an octal multiplication.

Example: Multiply 41_8 by 23_8.

Solution: Write

41_8 Factor
x 23_8 Factor
143 Partial Product
102 Partial Product
1163_8 Product

The hexadecimal multiplication table is given in figure 3-11 and the addition table in figure 3-7. As with the octal system, the best method to employ when performing multiplication is to use the appropriate tables.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	0	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	0	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	0	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	0	5	A	F	14	19	1D	23	28	2D	32	37	3C	41	46	4B
6	0	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	0	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	0	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	0	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	0	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	0	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	0	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	0	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	0	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	0	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

α, β, γ

Figure 3-11.—Hexadecimal multiplication table.

Example: Multiply $6C_{16}$ by 98_{16}.

Solution: Write

$6C_{16}$	Factor
98_{16}	Factor
360	Partial Product
3CC	Partial Product
4020_{16}	Product

Example: Multiply $BC2_{16}$ by EFA_{16}.

Solution: Write

$BC2_{16}$	Factor
x EFA_{16}	Factor
7594	Partial Product
B05E	Partial Product
A49C	Partial Product
$B01774_{16}$	Product

Division

The following terms are commonly used in division.

Dividend—a number to be divided.

Divisor—the number by which the dividend is divided.

Quotient—the number resulting from the division of one number by another.

Remainder—that which is left after subtraction or any deduction. (Division is a form of subtraction.)

Since division is the opposite of multiplication, the multiplication table

previously discussed may also be used for division. The division facts are as follows:

If A is the quotient, B the divisor, and C the dividend, then

$$C \div B = \frac{C}{B} = A, \text{ or } B\overline{)C}^{A} \quad B \neq 0$$

However, should A be the divisor, B the quotient, and C the dividend, then

$$C \div A = \frac{C}{A} = B, \text{ or } A\overline{)C}^{B} \quad A \neq 0$$

NOTE: As indicated above, division by 0 is not possible.

Use the information in figures 3-4 and 3-8 when performing division in the decimal system. The division facts are given in figure 3-8 and the subtraction facts in figure 3-4.

Example: Divide 54 by 9.

Solution: Write

$$\begin{array}{r} 6 \\ 9\,\overline{)54} \\ \underline{54} \\ 0 \end{array}$$

Example: Divide 252 by 6.

Solution: Write

$$\begin{array}{r} 42 \\ 6\,\overline{)252} \\ \underline{24} \\ 12 \\ \underline{12} \\ 0 \end{array}$$

The binary division facts are given in figure 3-9 and the subtraction facts in figure 3-5. As with multiplication, the process usually employed when performing division in a number system other than the decimal system is to convert the numbers to be divided (dividend and divisor) to their decimal equivalents, perform the indicated division, and then convert the resultant (quotient) to its equivalent in the number system in use. Again, this process is time consuming, and may lead to confusion and error. It should therefore be avoided and the appropriate tables be used.

Example: Divide 1111_2 by 11_2.

Solution: Write

$$\begin{array}{r} 101_2 \\ 11_2\,\overline{)1111_2} \\ \underline{11} \\ 011 \\ \underline{11} \\ 0 \end{array}$$

Example: Divide 101_2 by 10_2.

Solution: Write

$$\begin{array}{r} 10_2 \\ 10_2\,\overline{)101_2} \\ \underline{10} \\ 1 \quad \text{Remainder} \end{array}$$

Octal division facts are given in figure 3-10 and the subtraction facts in figure 3-6.

$$\begin{array}{r} 32_8 \\ 6_8\,\overline{)234_8} \\ \underline{22} \\ 14 \\ \underline{14} \\ 0 \end{array}$$

Example: Divide 765_8 by 4_8.

Solution: Write

$$\begin{array}{r} 175_8 \\ 4_8\,\overline{)765_8} \\ \underline{4} \\ 36 \\ \underline{34} \\ 25 \\ \underline{24} \\ 1 \quad \text{Remainder} \end{array}$$

Hexadecimal division facts are given in figure 3-11 and the subtraction facts in figure 3-7.

Example: Divide $D4E_{16}$ by 2_{16}.

Solution: Write

$$
\begin{array}{r}
6A7_{16} \\
2_{16}\ \overline{)\,D4E_{16}} \\
C \\
\hline
14 \\
14 \\
\hline
E \\
E \\
\hline
0
\end{array}
$$

Example: Divide $13BC_{16}$ by 3_{16}.

Solution: Write

$$
\begin{array}{r}
694_{16} \\
3_{16}\ \overline{)\,13BC_{16}} \\
12 \\
\hline
1B \\
1B \\
\hline
C \\
C \\
\hline
0
\end{array}
$$

CONVERSION TECHNIQUES

Two numbers written in different number systems that represent the same quantity are equivalent to each other. They have equal value even though the symbols or characters making up the numbers are entirely different. Therefore, it is possible to change or convert from one number system to another and still retain the value of the original number. The rest of this chapter describes several methods of converting integers and fractions from one number system to another. In the event a mixed number is encountered, the integer portion and the fractional portion must be converted separately.

The polynomial expansion method uses the positional notation process to derive the equivalent new number. A modification of polynomial expansion called the explosive method incorporates the digits of the original number during the conversion. The solution will be generated as a complete number. The digit-by-digit method generates the digits of the solution one at a time to be assembled for the solution. The grouping method provides a very fast method of grouping the digits to effect the conversion.

Polynomial Expansion

This method is an application of positional notation and uses the following rules for integer conversion:

A. Change the individual digits of the number being converted to a digit acceptable in the new system.

B. Change the old base to its new base equivalent.

C. Expand the number in accordance with the power series formula, using new base arithmetic.

D. Perform the indicated operations using new base arithmetic.

Q3-10. A decimal number is to be converted to hexadecimal by the polynomial expansion method. The number is expanded using the arithmetic of which number system?

Example: Convert 2378 to its octal equivalent.

A. Digit conversion

Decimal digits	2	3	7	8
Octal equivalents	2	3	7	10

B. Base (radix) conversion

$10_{10} = 12_8$

C. Expansion

$(2_8 \times 12_8^3) + (3_8 \times 12_8^2) + (7_8 \times 12_8^1) + (10_8 \times 12_8^0)$

A3-10. Hexadecimal

D. Perform indicated operations

$(2_8 \times 1750_8) + (3_8 \times 144_8) + (7_8 \times 12_8) + (10_8 \times 1_8) =$

$3720_8 + 454_8 + 106_8 + 10_8 =$

4512_8

Example: Convert 4512_8 to its decimal equivalent.

A. Digit conversion

Octal digits	4	5	1	2
Decimal equivalents	4	5	1	2

B. Base conversion

$10_8 = 8_{10}$

C. Expansion

$(4 \times 8^3) + (5 \times 8^2) + (1 \times 8^1) + (2 \times 8^0)$

D. Perform indicated operation

$(4 \times 512) + (5 \times 64) + (1 \times 8) + (2 \times 1) =$

$2048 + 320 + 8 + 2 =$

2378

Fractional conversions require two additional rules in the expansion step of the process.

Example: Convert .587 to its octal equivalent.

A. Digit conversion

Decimal digits	5	8	7
Octal equivalents	5	10	7

B. Base conversion

$10_{10} = 12_8$

C. Expansion—There are two additional steps. Together these steps remove the negative exponent and find the lowest common denominator.

(1) The negative powers of the base must be converted to positive powers, $x^{-1} = \frac{1}{x^1}$. Using this it is a simple matter to convert the number to a positive fraction. Therefore,

$(5_8 \times 12_8^{-1}) + (10_8 \times 12_8^{-2}) + (7_8 \times 12_8^{-3}) =$

$$\frac{5_8}{12_8^1} + \frac{10_8}{12_8^2} + \frac{7_8}{12_8^3} =$$

(2) The lowest common denominator must be found and the results of the expansion must be divided by the new base equivalent. By inspection it is found that 12_8^3 is the LCD. Therefore,

$$\frac{(5_8 \times 12_8^2) + (10_8 \times 12_8^1) + (7_8 \times 12_8^0)}{12_8^3} =$$

$$\frac{(5_8 \times 144_8) + (10_8 \times 12_8) + (7_8 \times 1_8)}{1750_8} =$$

$$\frac{764_8 + 120_8 + 7_8}{1750_8} = \frac{1113_8}{1750_8} = .45442_8+$$

NOTE: The + in the above answer indicates that the division could be carried out still further; however, this is a sufficient number of places for our purposes.

Example: Convert $.45442_8$ to its decimal equivalent.

A. Digit conversion

Octal digits	4	5	4	4	2
Decimal equivalents	4	5	4	4	2

B. Base conversion

$10_8 = 8_{10}$

C. Expansion

$$(4 \times 8^{-1}) + (5 \times 8^{-2}) + (4 \times 8^{-3}) + (4 \times 8^{-4}) + (2 \times 8^{-5}) =$$

$$\frac{(4 \times 8^4) + (5 \times 8^3) + (4 \times 8^2) + (4 \times 8^1) + (2 \times 8^0)}{8^5} =$$

$$\frac{(4 \times 4096) + (5 \times 512) + (4 \times 64) + (4 \times 8) + (2 \times 1)}{32768} =$$

$$\frac{(16384 + 2560 + 256 + 32 + 2)}{32768} = \frac{19234}{32768} = .5869+$$

When rounded off, the answer is .587, the original decimal fraction.

Keep in mind that polynomial expansion does not always provide an exact conversion when dealing with fractions. This occurs because certain numbers are not evenly divisible. For example, 2 cannot be divided by 3 and result in an exact answer.

The Explosive Method

When converting integers from one number system to another with the explosive method use the following rules:

A. Change the individual digits of the number being converted to a digit acceptable in the new system.

B. Change the old base to its new base equivalent.

C. Perform the following arithmetic operations using new base math.

1. Multiply the leftmost digit equivalent (MSD) by the new base equivalent of the old base.

2. Add to this product the next equivalent digit to the right and multiply this sum by the new base equivalent of the old base. Repeat step 2 until all digit equivalents of the number being converted have been used.

NOTE: DO NOT MULTIPLY AFTER THE LAST EQUIVALENT DIGIT ON THE RIGHT HAS BEEN ADDED.

Q3-11. What is the final step in convertin 769_{10} to the octal system using th explosive method?

Example: Convert 327 to its binary equivalent

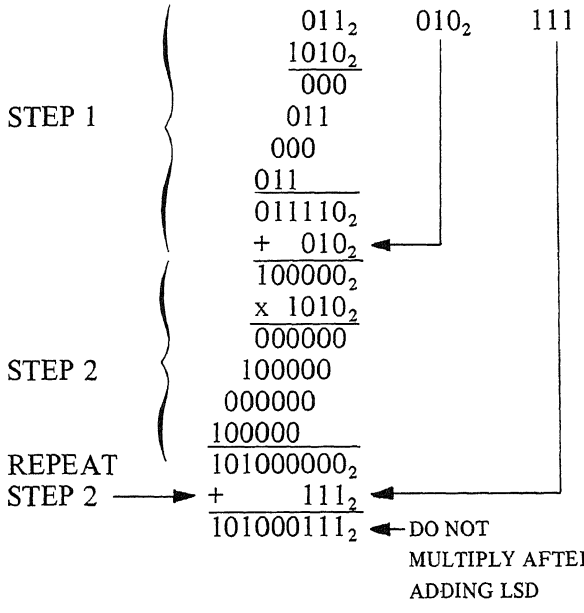
A. Digit conversion

Decimal digits	3	2	7
Binary equivalents	011	010	111

B. Base conversion

$10_{10} = 1010_2$

C. Perform arithmetic operation



Example: Convert 498 to its octal equivalent.

A. Digit conversion

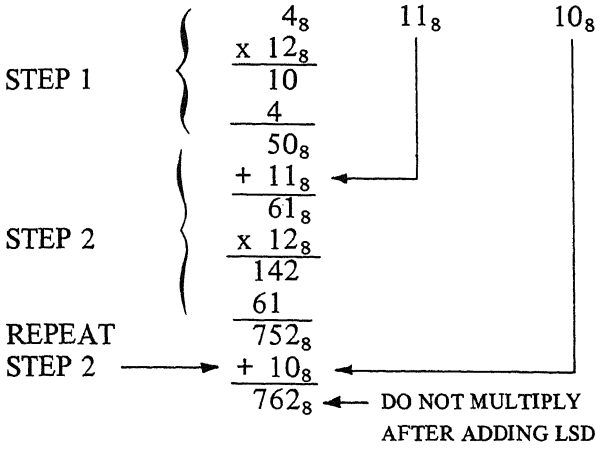
Decimal digits	4	9	8
Octal equivalents	4	11	10

B. Base conversion

$10_{10} = 12_8$

A3-11. Add the digit 9 to the previous product

C. Perform arithmetic operations



Example: Convert 762_8 to its hexadecimal equivalent.

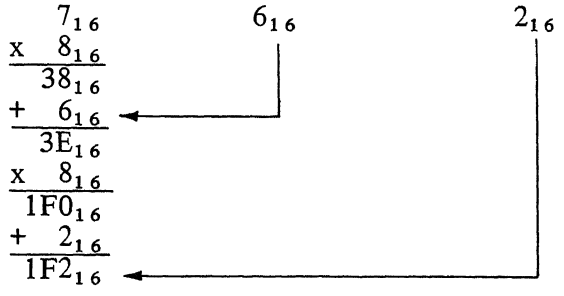
A. Digit conversion

Octal digits	7	6	2
Hexadecimal equivalents	7	6	2

B. Base conversion

$10_8 = 8_{16}$

C. Perform arithmetic operations



When converting fractions using the explosive method, the digit and base conversion rules are the same. The arithmetic steps are modified as follows:

(1) Divide the last digit equivalent on the right (LSD) by the new base equivalent of the old base.

(2) Add this quotient to the next digit equivalent on the left and divide this sum by the new base equivalent of the old base. Repeat step 2 until the sum which includes the last digit equivalent on the left (MSD) has been divided. Divide after the last digit has been added.

Example: Convert .987 to its octal equivalent.

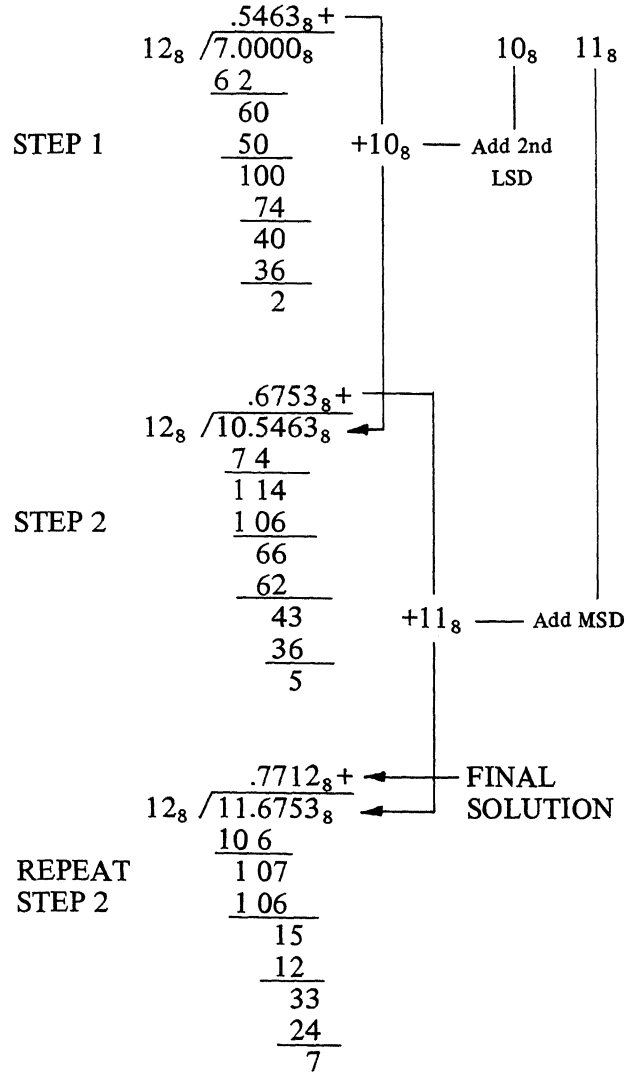
A. Digit conversion

Decimal digit	9	8	7
Octal equivalent	11	10	7

B. Base conversion

$10_{10} = 12_8$

C. Perform arithmetic operations



Note that the final division is made after the MSD is added into the dividend. Also, this method, like polynomial expansion, uses division and may not always provide an exact conversion of the fraction.

The Digit-by-Digit Method

This method provides an easy formula for converting from decimal numbers to other number systems. When converting integers, a process of repeated division is used observing the following rules:

A. Convert the new base to an old base equivalent.

B. Use old base math for the following arithmetic operations:

1. Divide the number to be converted by the old base equivalent of the new base; the remainder becomes the LSD of the answer and the quotient becomes the dividend for the next division.

2. Divide the previous quotient by the old base equivalent of the new base; the remainder becomes the next significant digit of the answer and the quotient becomes the dividend for the next division.

3. Continue this process until a quotient of zero (0) is obtained. The remainder of this final division will be the MSD of the answer.

4. Where necessary, convert the remainders to a form acceptable in the new base.

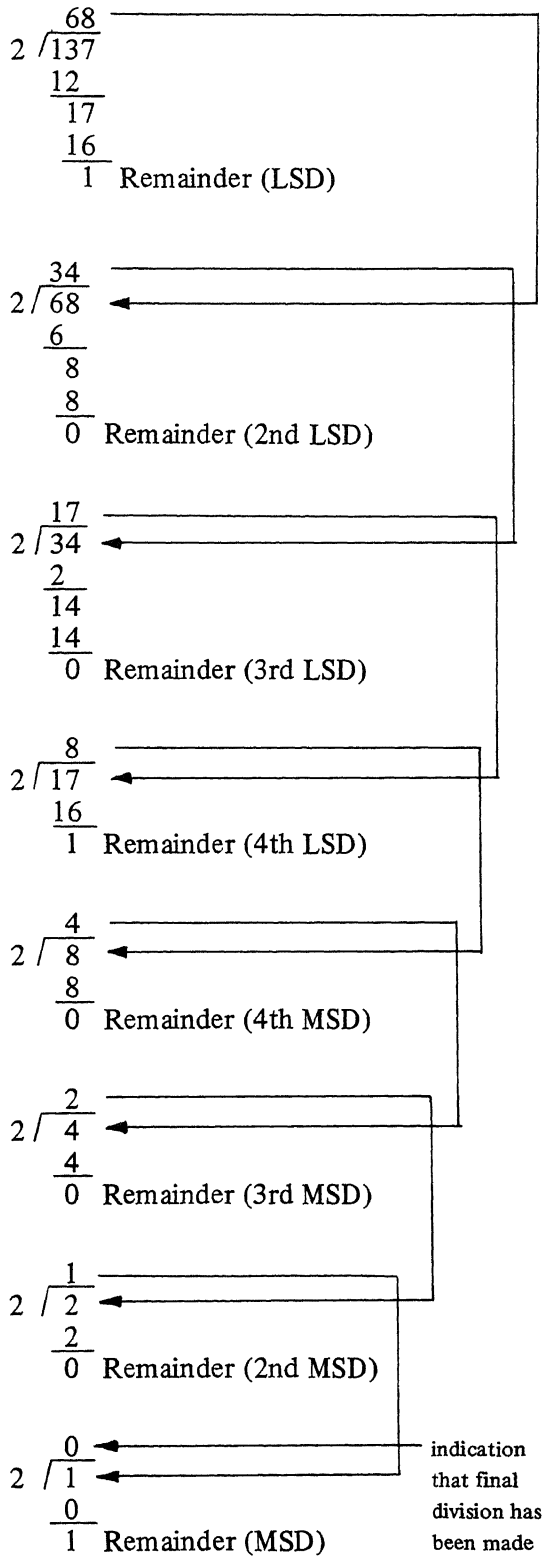
Q3-12. What number system is used for the arithmetic operations in the digit-by-digit method of conversion?

Example: Convert 137 to its binary equivalent.

A. Base conversion

$10_2 = 2_{10}$

B. Perform arithmetic operations



A3-12. Old base.

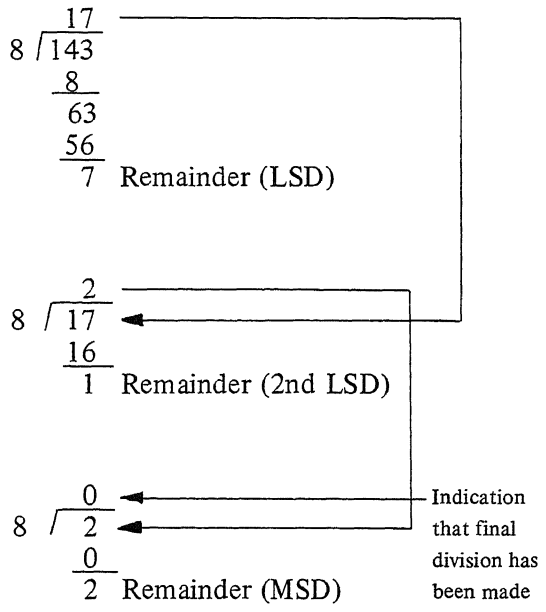
Arranging the remainders in their proper sequence or order gives 10001001_2, which is the binary equivalent of 137_{10}.

Example: Convert 143 to its octal equivalent.

A. Base conversion

$10_8 = 8_{10}$

B. Perform arithmetic operations



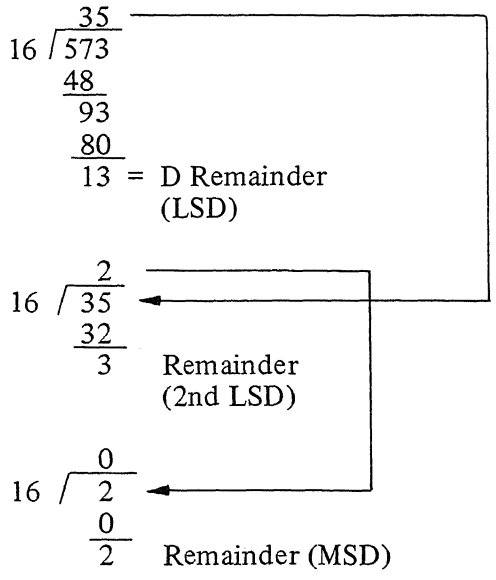
Arranging the remainders in their proper sequence or order gives 217_8, which is the octal equivalent of 143_{10}.

Example: Convert 573 to its hexadecimal equivalent.

A. Base conversion

$10_{16} = 16_{10}$

B. Perform arithmetic operations



Arranging the remainders in their proper sequence gives $23D_{16}$, which is the hexadecimal equivalent of 573_{10}.

When converting fractions, the digit-by-digit method uses a process of repeated multiplication observing the following rules:

A. Convert the new base to an old base equivalent.

B. Use old base math for the following arithmetic operations:

1. Multiply the fraction to be converted by the old base equivalent of the new base. The integer portion of the product becomes the MSD of the answer, and the fractional portion becomes the multiplicand for the next multiplication.

2. Multiply the fractional portion of the previous product by the old base equivalent of the new base. The integer portion of this product becomes the next significant digit of the answer, and the fractional portion becomes the multiplicand for the next multiplication.

3. Continue this process until the desired degree of accuracy is attained.

4. Where necessary, convert the integer portion of the products to a form acceptable in the new base.

Example: Convert .425 to its binary equivalent.

A. Base conversion

$10_2 = 2_{10}$

B. Perform arithmetic operations

	.425	
	x 2	
MSD →	0.850	.0
	x 2	
2nd MSD →	1.700	.01
	x 2	
3rd MSD →	1.400	.001
	x 2	
4th MSD →	0.800	.0000
	x 2	
5th MSD →	1.600	.00001
	x 2	
6th MSD →	1.200	.000001
	x 2	
7th MSD →	0.400	.0000000
	x 2	
8th MSD →	0.800	.00000000
	x 2	
9th MSD →	1.600	.000000001
	x 2	
LSD →	1.200	.0000000001
		$.0110110011_2$

Example: Convert .589 to its octal equivalent.

A. Base conversion

$10_8 = 8_{10}$

B. Perform arithmetic operations

	.589	
	x 8	
MSD →	4.712	.4
	x 8	
2nd MSD →	5.696	.05
	x 8	
3rd MSD →	5.568	.005
	x 8	
LSD →	4.544	.0004
		$.4554_8$

Example: Convert .4563 to its hexadecimal equivalent.

A. Base conversion

$10_{16} = 16_{10}$

B. Perform arithmetic operations

.4563	
x 16	
27378	
4563	
7.3008	.7
x 16	
18048	
3008	
4.8128	.04
x 16	
48768	
8128	
13.0048	.00D
x 16	
288	
48	
0.0768	.0000
x 16	
4608	
768	
1.2288	.00001
	$.74D01_{16}$ Solution

Grouping

Generally speaking, the grouping method of conversion may be used when the radix of the higher order number system is an integral power of the radix of the lower order number system. (For example, eight is an integral power of two.) This method is particularly useful when converting between bases 2, 8, and 16. A variation of this method can also be effectively used for conversion to the decimal system from any of the three previously mentioned number systems.

The procedure to follow when converting from binary to octal is to start at the radix point and, working in both directions, divide the binary number into groups of three digits (filling in with zeros whenever a group is incomplete).

After the binary number has been properly divided into groups, each group is assigned its octal equivalent. The resulting number will be the octal equivalent of the binary number.

Q3-13. Binary to octal conversion can be performed by grouping the binary digits into units of_____.

Example: Convert 1101111.1101_2 to its octal equivalent.

Original number

1101111.1101_2

Grouping

1 101 111.110 1

Adding the necessary zeros

001 101 111.110 100

Assigning the octal equivalents

001 101 111.110 100

1 5 7 6 4

$1101111.1101_2 = 157.64_8$

To convert from octal to binary, the process s reversed. Maintaining the order or sequence of he original number, assign each digit of the ıumber its binary equivalent. The resulting ıumber will be the binary equivalent of the ɔctal number.

Example: Convert 1507.06_8 to its binary equivalent.

Original number

1507.06_8

Assigning the binary equivalents

1 5 0 7 . 0 6

001 101 000 111.000 110

$1507.06_8 = 001101000111.000110_2$

Since here they are of no significance, the zeros at the beginning and end of the binary equivalent are usually omitted. The binary equivalent of 1507.06_8 may be written as 1101000111.00011_2.

Sixteen is also an integral power of two and, as previously indicated, grouping may be used when converting from binary to hexadecimal and vice versa. With one exception the process is the same as used for binary/octal conversions. The exception is that when converting from binary to hexadecimal, the binary number is divided into groups of <u>four</u> digits instead of three. Each group is then assigned its hexadecimal equivalent.

Example: Convert 111011.01101_2 to its hexadecimal equivalent.

Original number

111011.01101_2

Grouping

11 1011.0110 1

Adding the necessary zeros

0011 1011.0110 1000

Assigning the hexadecimal equivalent

0011 1011.0110 1000

3 B . 6 8

$111011.01101_2 = 3B.68_{16}$

When converting from hexadecimal to binary, each hexadecimal digit is assigned its binary equivalent.

Example: Convert $3C8.96_{16}$ to its binary equivalent.

Original number

$3C8.96_{16}$

Assigning the binary equivalents

3 C 8 . 9 6

0011 1100 1000.1001 0110

As before, the zeros at the beginning and end of the binary equivalent have no significance and may be omitted. Thus,

$3C8.96_{16} = 1111001000.1001011_2$

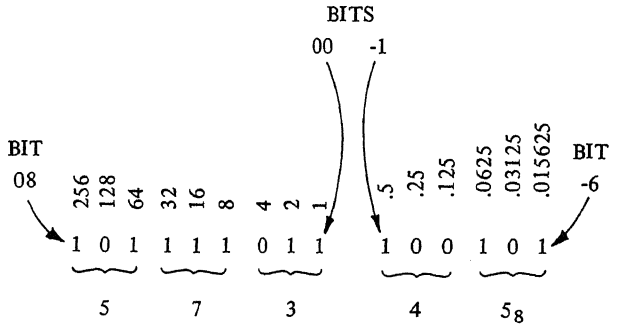
BIT POSITIONS.—Each binary digit or bit has a decimal value associated with its position within the number. Immediately to the left of the radix point is the first bit position (bit 00). (See fig. 3-12.) It has a weighting value of 1. Next to the left is bit 01 with a value of 2. Each succeeding bit position to the left has a decimal value twice that of the immediately preceding bit position. It follows, then, that when moving to the right each bit position will have a value one-half that of the bit position immediately to its left. A partial table of these weighting values and associated bit positions (called the positive and negative powers of two) is shown in figure 3-12. Note that beginning at the radix point the integer values are to the left and the fractional values are to the right.

A hexadecimal or octal number can be converted to the decimal number system by first using the grouping method to indicate the binary representation of the original number. For example, 573.45_8 is equivalent to the binary number 101111011.100101_2.

$$5 \quad 7 \quad 3 \,.\, 4 \quad 5_8$$

$$101 \quad 111 \quad 011.100 \quad 101_2$$

All that is required to convert a binary number to a decimal is to add up the decimal weighting values of the bit positions occupied by a 1. Using the preceding example—



A tabulation of the decimal values of the bit positions shown above, occupied by a 1 (do not add in the bit positions with a 0), shows that the decimal equivalent of 573.45_8 is 379.578125.

Tabulating bits:

256 ——— Bit 08
64
32
16
8
2
1
.5
.0625
.015625 — Bit -6
379.578125

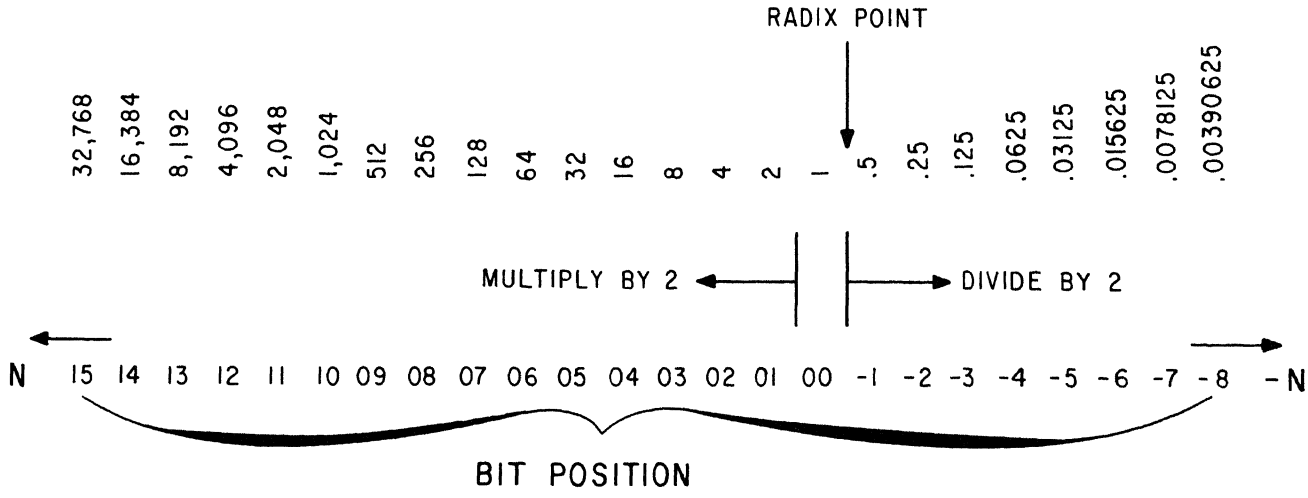
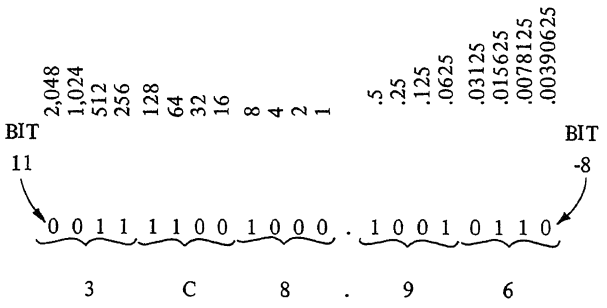


Figure 3-12.—Partial table of the positive and negative powers of two.

A3-13. three digits

When converting a base 16 number with this method, remember that each digit requires four binary bits to represent it. For example, the hexadecimal number 3C8.96 is equivalent to 0011 1100 1000.1001 0110_2. Laying out the decimal weighting values of the individual bits results in



Tabulating the decimal values of the bit positions with a 1 results in a decimal equivalent of 968.5859375.

Tabulating bits:

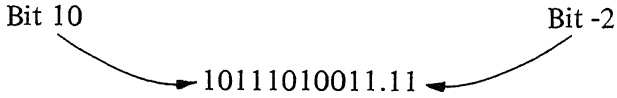
```
   512
   256
   128
    64
     8
    .5
    .0625
    .015625
   .0078125
968.5859375
```

Decimal to octal or hexadecimal conversions are a reasonably simple process with this method. From the decimal number, subtract the largest bit weighting value that will leave a positive or zero result. Write a 1 in that bit position. If the result was not equal to zero, subtract the next bit weighting value that will leave a positive or zero result and write a 1 in that bit position. Continue this process until a zero result is obtained. Then make the direct conversion from binary to octal or hexadecimal by the grouping method.

Example: Convert 1491.625 to its octal and hexadecimal equivalents.

1491.625		
- 1024	Bit 10 value	Write 1 in these bit positions and zero in all others
467.625		
- 256	Bit 8 value	
211.625		
- 128	Bit 7 value	
83.625		
- 64	Bit 6 value	
19.625		
- 16	Bit 4 value	
3.625		
- 2	Bit 1 value	
1.625		
- 1	Bit 0 value	
0.625		
- 0.5	Bit -1 value	
0.125		
- 0.125	Bit -2 value	
0.0		

Writing a 1 in each bit position where the weighting value was used and 0 in all others results in the binary number.



Using the grouping method previously discussed shows that the octal and hexadecimal equivalents are

$$5\ D\ 3\ .\ C_{16}$$

$$010111010011.1100$$

$$2\ 7\ 2\ 3\ .\ 6_8$$

Insignificant zeros have been added on both ends of the binary number to ensure enough bit positions to properly convert the number.

Note that the only difference in converting to either octal or to hexadecimal is the grouping of the bits, three for octal or four for hexadecimal.

SUMMARY

What have you learned in these three chapters? First, the information about logic computation described basically how to manipulate the variables used with Boolean algebra. If the state of a variable was defined as A = true, then the opposite state or $\overline{A}$ was defined as false. Most important was the fact that a variable can assume only one of two possible conditions at any given time.

The arithmetic operators (•) AND and (+) OR were also discussed. An AND function requires that ALL variables be in their defined state for the function to be true. In an OR function, any one of the variables in its defined state is enough to satisfy the function.

Next came a discussion of logic polarity, which explained the differences between positive and negative logic. Positive logic needs a relatively more positive signal to activate a logic circuit. If a 1 is +5 volts, then a 0 must be some voltage level relatively more negative or possibly +2 volts. Negative logic, the opposite of positive logic, requires that the activating input to a logic circuit be of a relatively more negative voltage level. For instance, if a 1 is +5 volts, then a 0 signal could be +10 volts. The currently used symbology for indicating a positive or negative logic circuit was also discussed. A small flag indicates negative logic and no flag indicates positive logic.

In the discussion of logic circuits the most common circuits were emphasized. The AND, OR, Inverter or Negator, NAND, NOR, and Exclusive OR circuits were illustrated and explained. Truth tables, waveforms, circuit schematics, and current symbology were included in the discussion. The Inverter circuit complements its input. A complemented AND output is produced by the NAND circuit, while the NOR circuit output is a complemented OR function. The Exclusive OR circuit will output a true when any one input is true and the other is false. It will output a false when all inputs are true or all the inputs are false. As for the symbology, keep in mind the difference between negative and positive logic. The functional output of the circuit remains the same whether positive or negative logic is implemented.

In the discussion of Boolean algebra, the basic rules, laws, theorems, and axioms were described and illustrated. If you are using this chapter because of the demands of your job, you should memorize these principles. Techniques for converting a given logic equation into a logic diagram and developing a logic equation from an existing logic diagram were discussed. Certain simplification techniques including the use of the basic principles of Boolean algebra, the Vietch diagram, and Harvard charts were also a part of this chapter. These techniques were included because they will be used when troubleshooting digital equipments.

The final portion of the publication dealt with the various number systems used in Navy digital computers. Octal, binary, and hexadecimal systems were described in detail. Methods of conversion between each of these systems and the decimal system were explained as were the fundamental arithmetic operations of addition, subtraction, multiplication, and division.

APPENDIX I

GLOSSARY

ABSORPTION, LAW OF: In Boolean algebra, the law which states that the odd term will be absorbed when a term is combined by logical multiplication with the logical sum of the term and another term, or when a term is combined by logical addition with the logical product of the term and another term.

ACCURACY: The quality of freedom from mistake or error; that is, the degree of conformity to truth or to a rule.

ADDEND: The quantity that is added to another quantity (called the augend) to produce a sum.

ANALYSIS: The methodical investigation of a problem, and the separation of the problem into smaller related units for further detailed study.

AND: A logic operator having the property that if P is a statement, Q is a statement, R is a statement . . . then the AND of P, Q, R . . . , is true if ALL statements are true, false if ANY statement is false.

AND GATE: An electronic gate whose output is active only when every input is in its active state. An AND gate performs the function of the logical "AND." Also called AND-circuit.

ANODE: The positive electrode of an electrochemical device, such as a primary or secondary cell, toward which negative ions are drawn.

ARITHMETIC UNIT: The unit of a computing system that contains the circuits that perform arithmetic operations.

AUGEND: In arithmetic addition, a number that increases when another number (called the addend) is added to it.

AXIOM: A statement regarded as a self-evident truth.

BASE: (1) A reference value. (2) A number that is multiplied by itself as many times as indicated by an exponent. (3) Same as radix.

BASE NUMBER: The radix of a number system (10 is the radix, or base number, for the decimal system; 8 is the base number for the octal system).

BIAS: (1) An electrical force applied to a relay, vacuum tube, or semiconductor to establish an electrical reference level for the operation of the device. (2) The d.c. potential applied between elements of a transistor to make the device perform in the desired manner.

BIAS CURRENT: Current which flows through the base-emitter junction of a transistor. It can be adjusted to set the operating point of the transistor.

BINARY: (1) Pertaining to a characteristic or property involving a selection, choice, or condition in which there are at most two possibilities; (2) Pertaining to the number representation system with a radix of two.

BINARY CODE: A code that makes use of exactly two distinct characters, usually 0 and 1.

BINARY DIGIT: In binary notation, either of the characters 0 or 1.

BINARY NOTATION: Fixed radix notation where the radix is two.

BIT: A binary digit; for example, zero or one. It may be equivalent to an "on" or "off" condition.

BOOLEAN ALGEBRA: Mathematical logic that deals with classes, off-on circuit elements, and propositions. Uses logic gates such as AND, OR, NOT, etc. Introduced in 1847 by English mathematician George Boole.

CARRY: One or more digits, produced in connection with an arithmetic operation on one digit place of two or more numerals in positional notation, that are forwarded to another digit place for processing there.

CHARACTER: A letter, digit, or other symbol that is used as part of the organization, control, or representation of data.

CHECK: A process for determining accuracy.

COLLECTOR: (1) In a transistor, primary current flows through this electrode. (2) The external terminal of a transistor that is connected to this region.

COMPLEMENT: A number that can be derived from a specified number by subtracting it from a second specified number. For example, in radix notation, the second specified number may be a given power of the radix or one less than a given power of the radix. The negative of a number is often represented by its complement.

COMPLEMENTARY LAW: In Boolean algebra, this law states that the logical addition of a quantity and its complement will result in 1, and the logical multiplication of a quantity and its complement will result in a product of 0.

COMPUTER: A data processor that can perform substantial computation, including numerous arithmetic or logic operations, without intervention by a human operator during the run.

COMMUTATIVE LAW: In Boolean algebra this law states that changing the order of the terms in an equation will not affect the value of the equation.

Example: $A + B = B + A$; $A \cdot B = B \cdot A$

CUTOFF: The condition of a transistor when zero or a reverse bias is applied to the emitter-base junction and collector current ceases to flow.

DECIMAL: Pertaining to the number representation system with a radix of ten.

DECIMAL DIGIT: In the decimal number system, one of the characters 0 through 9.

DECIMAL NOTATION: A fixed radix notation where the radix is ten.

DECIMAL NUMERAL: A decimal representation of a number.

DECIMAL POINT: The radix point in decimal representation.

DEMORGAN'S THEOREM: A theorem which states that the negation or inversion of an expression that is ANDed is equal to the same expression of inverted OR implications, or the negation or inversion of an expression that is ORed is equal to the same expression of inverted AND implications. In symbols,

$$\overline{XYZ} = \overline{X} + \overline{Y} + \overline{Z} \text{ or } \overline{X + Y + Z} = \overline{X} \cdot \overline{Y} \cdot \overline{Z}$$

DIGIT: A symbol that represents one of the positive integers smaller than the radix.

DIGITAL: Pertaining to data in the form of digits.

DIGITAL COMPUTER: (1) A computer in which discrete representation of data is mainly used. (2) A computer that operates on discrete data by performing arithmetic and logic processes on these data.

DIGITIZE: To use numeric characters to express or represent data.

DISTRIBUTIVE LAW: In Boolean algebra, the law which states that if a group of terms connected by like operators contains the same variable, the variable may be removed from the terms and associated with them by the appropriate sign of operation.

DIVIDEND: The number that is divided by another number.

DIVISOR: The number by which the dividend is divided.

DOUBLE NEGATION, LAW OF: In Boolean algebra, the law which states that the complement of the complement of a term is the equivalent of the term.

DUALIZATION, LAW OF: See DEMORGAN'S THEOREM.

DUODECIMAL: The number system with a radix of twelve.

EMITTER: (1) An element in a transistor that sends current carriers into the base of the transistor. (2) In a vacuum tube, the cathode.

EXCLUSIVE OR: A logical operator which has the property that if A and B are two logic statements, then the statement $A \oplus B$, where the $\oplus$ is the Exclusive OR operator, is true if either A or B, but not both are true, and false if A and B are both false or both true. The boolean expression for the Exclusive OR operation is $f = A\overline{B} + \overline{A}B$.

EXCLUSIVE-OR CIRCUIT: A circuit that produces an active output signal when any one, but not more than one, input is in its active state.

EXPONENT: In a floating point representation, the numeral written in superscript (10^2), representing a number that indicates the power to which the base is to be raised.

EXPRESSION: A series of variables that are connected by operating symbols to describe a desired computation.

FACTOR: Any of the numbers, quantities, or symbols which, when multiplied together, form a product.

FLIP-FLOP: A device having two stable states and two inputs (or types of input signals), each of which corresponds with one of the two states. The circuit remains in either state until caused to change to the other state by application of a pulse.

FORWARD BIAS: A bias voltage applied to a semiconductor junction with polarity such that the junction is activated.

GATE: As applied to logic circuitry, one of several different types of electronics devices that will provide a particular output when specified input conditions are satisfied. (AND, OR, Inverter)

GATING: The application of a particular waveform used to perform electronic switching.

GROUND: The point in a circuit used as a common reference point for measuring purposes.

HARVARD CHART: In terms of logic expression simplification, lists all the possible variable combinations under consideration and the complements of these variable combinations.

HEXADECIMAL: A number system with a base of sixteen.

HEXADECIMAL SYSTEM: Pertaining to the number system with a radix of sixteen. (It uses the ten digits of the decimal system and the first six letters of the alphabet.)

IDENTITY, LAW OF: In Boolean algebra, the law which states that, when combining three or more terms, in either logical addition or logical multiplication, the order in which the terms are combined will not affect the result.

IDEMPOTENT LAW: In Boolean algebra, combining a quantity with itself either by logical addition or logical multiplication will result in a logical sum or product that is the equivalent of the quantity. Example: $A + A = A$; $A \cdot A = A$

INHIBIT: To prevent the occurrence of an event.

INPUT: The current, voltage, power, or activating force applied to a circuit or device.

INPUT/OUTPUT: (I/O) Pertaining to either input or output or both.

INSTRUCTION: A statement that specifies an operation and the values or locations of its operands.

INVERTER CIRCUIT: A circuit which performs the NOT operation. With an input of A the output is $\overline{A}$, and with an input of $\overline{A}$ the output is A.

LOGIC: The basic principles and applications of truth tables, interconnections of off-on circuit elements, and other factors involved in mathematical computation in automatic data processing systems and other devices.

LOGIC CIRCUIT: The primary units in a digital equipment, made up of electronic gates.

LOGIC DIAGRAM: In computers and data processing equipment, a diagram representing the logical elements and their interconnections.

LOGIC ELEMENT: A device that performs the logic function. The smallest building blocks which can be represented by operators in an appropriate system.

LOGIC INSTRUCTION: Any instruction that executes an operation that is defined in symbolic logic, such as AND, OR, NOR.

LOGIC OPERATION: A non-arithmetical operation in a computer, where logical YES or NO quantities are involved.

LOGIC SWITCH: A diode matrix or other switching arrangement that is capable of directing an input signal to one of several outputs.

LOGIC SYMBOL: A symbol used to represent a logic element graphically. Also a symbol used to represent a logic operator.

MATRIX: In computers, a logic network in the form of an array of input leads and output leads with logic elements connected at some of their intersections.

MINUEND: The number from which another number is to be subtracted.

MISTAKE: A human action that produces an unintended result.

MODULUS: The total number of different numbers or stable conditions that a counting device can indicate.

MULTIPLICAND: The number that is to be multiplied by another number (called the multiplier).

MULTIPLIER: The number by which another number is multiplied.

NAND GATE: Basically, the circuit operates by ANDing a number of logic signals together; it then uses its output signals as the input to Inverter circuit which complements or negates the results.

NEGATION: The process of inverting or complementing the value of a function or variable.

NEGATIVE LOGIC: When the signal that activates the circuit (a 1, high, or true) has an electrical level that is relatively more NEGATIVE than the other logic state, the logic polarity is considered to be NEGATIVE.

NOR GATE: The NOR gate combines the functions of an OR gate and the Inverter circuit. It produces a circuit in which the output is a logic 1 only if ALL inputs are a logic 0.

NOT: A logic function having the property that it inverts its input to provide the opposite output.

NUMBER: A symbol (called a numeral), or group of symbols, representing a sum of units.

NUMBER SYSTEM: Any set of symbols or characters used for the purpose of counting objects and performing arithmetic operations.

OCTAL (NUMBERING SYSTEM): A numbering system with a radix of 8.

PARTIAL PRODUCT: In mathematics, the intermediate sum obtained when the multiplicand is multiplied by each digit in a multiplier having more than one digit.

POLYNOMIAL EXPANSION: Method which uses the positional notation process to derive the equivalent new number. It includes changing the individual digits of the number being converted to digits acceptable in the new system and changing the old base to the new base equivalent.

POSITIONAL NOTATION: See POSITIONAL VALUE.

POSITIONAL VALUE: The principle of positional value consists of assigning a digit a value which depends on two factors: (1) the digit's basic value, that is, the number of units it represents by itself; and (2) a weighting value, which is determined by the digit's position within a given number.

POSITIONAL WEIGHTING: The value given a digit based on the digit's position within a given number.

POSITIVE LOGIC: When the signal that activates the circuit (a 1, high, or true) has an electrical level that is relatively more POSITIVE than the other logic state, the logic polarity is considered to be POSITIVE.

POSTULATES: In mathematics or logic, an axiom or hypothesis.

PRODUCT: The number resulting from the multiplication of two or more numbers.

QUANTITY: In mathematics it designates a positive or negative number.

QUOTIENT: The number which results when one number is divided by another.

RADIX (OR BASE): The number of symbols a number system uses including zero. The value of the radix is always one unit greater than the largest basic character being used.

REGISTER: A number of flip-flops arranged in a chain.

REMAINDER: In division, the number left after the quotient has been found which cannot be divided by the divisor without resulting in a fraction.

REVERSE BIAS: An external voltage applied to a diode or semiconductor junction to reduce the flow of current across the junction. (Also called back bias.)

SUBTRAHEND: The number which is subtracted from the minuend.

SUM: The results of an addition.

THEOREM: A formula or statement in mathematics or logic which is based on or deduced from other formulas or statements.

TRUTH TABLE: A table that describes a logic function by listing all possible combinations of input values and indicating, for each combination, the true output values.

UNIT: A single object or thing.

V_{CC}: Collector supply voltage.

VARIABLE: A representative symbol that can assume any of a given set of values.

VEITCH DIAGRAM: A diagram used to find the simplest logic equation needed to express a given function.

VINCULUM: A straight horizontal line placed over terms of an expression; it serves the same grouping purpose as parenthesis and brackets and indicates negation.

INDEX